Idéalisation d'assemblages CAO pour l'analyse EF de structures

Flavien Boussuge

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématiques et Informatique**

Arrêté ministérial : 7 août 2006

Présentée par

Flavien Boussuge

Thèse dirigée par **Jean-Claude Léon**
et codirigée par **Stefanie Hahmann**

préparée au sein **Laboratoire Jean Kuntzmann - INRIA Grenoble, et AIRBUS Group Innovations**
et de **l'École Doctorale Matématiques, Sciences et Technologies de l'Information, Informatique**

Idealization of CAD assemblies for FE structural analyses

Idéalisation d'assemblages CAO pour l'analyse EF de structures

Thèse soutenue publiquement le **TBD**,
devant le jury composé de :

Prof., Cecil Armstrong
Queen's University Belfast, Rapporteur
Dr., Bruno Lévy
Directeur de recherche, INRIA Nancy, Rapporteur
Dr., Lionel Fine
AIRBUS Group Innovations, Suresnes, Examinateur
Prof., Jean-Philippe Pernot
Arts et Métiers ParisTech, Aix en Provence, Examinateur
Prof., Jean-Claude Léon
INP-Grenoble, ENSE3, Directeur de thèse
Prof., Stefanie Hahmann
INP-Grenoble, ENSIMAG, Co-Directeur de thèse

M., Nicolas Chevassus
AIRBUS Group Innovations, Suresnes, Invité
M., François Guillaume
AIRBUS Group Innovations, Suresnes, Invité

The research described in this thesis was carried out at the Laboratoire Jean Kuntzmann LJK-INRIA research team Imagine and the SPID team of AIRBUS Group Innovations.

This work was supported by a CIFRE convention of the ANRT and the French Ministry of Higher Education and Research.

Idealization of CAD assembly for FE structural analysis

Abstract

Aeronautical companies face a significant increase in complexity and size of simulation models especially at the level of assemblies, sub-systems of their complex products. Pre-processing of Computer Aided Design (CAD) models derived from the digital representation of sub-systems, i.e., Digital Mock-Ups (DMUs), into Finite Elements Analysis (FEA) models requires usually many tedious manual tasks of model preparation and shape transformations, in particular when idealizations of components or assemblies have to be produced. Therefore, the purpose of this thesis is to make a contribution to the robust automation of the time-consuming sequences of assembly preparation processes.

Starting from an enriched DMU with geometrical interfaces between components and functional properties, the proposed approach takes DMU enrichment to the next level by structuring components' shapes. This approach extracts a construction graph from B-Rep CAD models so that the corresponding generative processes provide meaningful volume primitives for idealization application. These primitives form the basis of a morphological analysis which identifies the sub-domains for idealization in the components' shapes and their associated geometric interfaces. Subsequently, models of components as well as their geometric representation get structured in an enriched DMU which is contextualized for FEA application.

Based on this enriched DMU, simulation objectives can be used to specify geometric operators that can be robustly applied to automate components and interfaces shape transformations during an assembly preparation process. A new idealization process of standalone components is proposed while benefiting from the decomposition into sub-domains and their geometric interfaces provided by the morphological analysis of the component. Interfaces between sub-domains are evaluated to robustly process the connections between the idealized sub-domains leading to the complete idealization of the component.

Finally, the scope of the idealization process is extended to shape transformations at the assembly level and evolves toward a methodology of assembly pre-processing. This methodology aims at exploiting the functional information of the assembly and interfaces between components to perform transformations of groups of components and assembly idealizations. In order to prove the applicability of the proposed methodology, corresponding operators are developed and successfully tested on industrial use-cases.

Keywords : Assembly, DMU, idealization, CAD-CAE integration, B-Rep model, generative shape process, morphological analysis

Idéalisation d'assemblages CAO pour l'analyse EF de structures

Résumé

Les entreprises aéronautiques ont un besoin continu de générer de grands et complexes modèles de simulation, en particulier pour simuler le comportement structurel de sous-systèmes de leurs produits. Actuellement, le pré-traitement des modèles de Conception Assistée par Ordinateur (CAO) issus des maquettes numériques de ces sous-systèmes en Modèles Eléments Finis (MEF), est une tâche qui demande de longues heures de travail de la part des ingénieurs de simulation, surtout lorsque des idéalisations géométriques sont nécessaires. L'objectif de ce travail de thèse consiste à définir les principes et les opérateurs constituant la chaîne numérique qui permettra, à partir de maquettes numériques complexes, de produire des géométries directement utilisables pour la génération de maillages éléments finis d'une simulation mécanique.

A partir d'une maquette numérique enrichie d'information sur les interfaces géométriques entre composants et d'information sur les propriétés fonctionnelles de l'assemblage, l'approche proposée dans ce manuscrit est d'ajouter un niveau supplémentaire d'enrichissement en fournissant une représentation structurelle de haut niveau de la forme des composants CAO. Le principe de cet enrichissement est d'extraire un graphe de construction de modèles CAO B-Rep de sorte que les processus de génération de forme correspondants fournissent des primitives volumiques directement adaptées à un processus d'idéalisation. Ces primitives constituent la base d'une analyse morphologique qui identifie dans les formes des composants à la fois des sous-domaines candidats à l'idéalisation mais également les interfaces géométriques qui leurs sont associées. Ainsi, les modèles de composants et leurs représentations géométriques sont structurés. Ils sont intégrés dans la maquette numérique enrichie qui est ainsi contextualisée pour la simulation par EF.

De cette maquette numérique enrichie, les objectifs de simulation peuvent être utilisés pour spécifier les opérateurs géométriques adaptant les composants et leurs interfaces lors de processus automatiques de préparation d'assemblages. Ainsi, un nouveau procédé d'idéalisation de composant unitaire est proposé. Il bénéficie de l'analyse morphologique faite sur le composant lui fournissant une décomposition en sous-domaines idéalisables et en interfaces. Cette décomposition est utilisée pour générer les modèles idéalisés de ces sous-domaines et les connecter à partir de l'analyse de leurs interfaces, ce qui conduit à l'idéalisation complète du composant.

Enfin, le processus d'idéalisation est étendu au niveau de l'assemblage et évolue vers une méthodologie de pré-traitement automatique de maquettes numériques. Cette méthodologie vise à exploiter l'information fonctionnelle de l'assemblage et les informations morphologiques des composants afin de transformer à la fois des groupes de

composants associés à une même fonction ainsi que de traiter les transformations d'idéalisation de l'assemblage. Pour démontrer la validité de la méthodologie, des opérateurs géométriques sont développés et testés sur des cas d'application industriels.

Mots-clés : Assemblage, Maquette Numérique, intégration CAO-calcul, modèle B-Rep, graphe de construction , processus génératif de forme, idéalisation

Table of contents

List of Figures

List of Tables

Acronyms

B-Rep Boundary representation. 10–14, 17, 20, 21, 34, 45, 49, 50, 52, 62, 63, 77, 78, 85, 86, 89, 91–93, 96–98, 101, 102, 114, 116–118, 120, 140, 141, 147, 171, 192–195

CAD Computer Aided Design. IX, 2, 3, 5–7, 9–14, 16–19, 21, 24–26, 29, 30, 33–36, 39–41, 43, 44, 47–53, 55, 59, 60, 63, 65, 67, 70, 73, 77–79, 86–93, 97–99, 114, 116, 118–120, 122, 123, 140, 144, 155, 156, 159, 169, 179, 180, 183, 186, 189, 191, 193, 195

CAD-FEA Computer Aided Design to Finite Element Analysis(es). 12, 43, 191–193

CAE Computer Aided Engineering. 11, 12, 23, 25, 26, 33, 35, 37–41, 43–45, 47, 50–53, 67, 77, 123, 156, 165, 180, 189

CSG Constructive Solid Geometry. 9–11, 13, 62, 63

DMU Digital Mock-Up. XIX, 2, 3, 5–8, 11, 12, 16–21, 23, 30–35, 37–41, 43, 64, 67–75, 77–79, 82, 83, 85, 86, 142, 157, 159, 160, 167, 168, 171, 172, 178–180, 183, 189, 191–193, 196

FE Finite Elements. 5, 7, 22, 25, 27–32, 38–41, 44–47, 49, 52, 54, 64, 70, 72, 77, 79, 82, 85, 121–123, 125, 136, 145, 152, 153, 157, 159, 162, 165, 169, 177, 180, 183, 185, 186, 189, 191, 193

FEA Finite Element Analysis(es). 1–3, 5, 7, 12, 22, 28, 30–32, 40, 43, 45, 46, 49, 57, 58, 60, 64, 67–70, 72, 74–79, 83, 85, 87–89, 96, 117, 119–121, 123, 141, 156, 160, 168, 172, 173, 176, 177, 180, 191

FEM Finite Element Models. 1, 22–24, 26, 27, 29, 44, 46, 69–72, 76, 79, 82, 88, 121, 147, 183

KBE Knowledge Based Engineering. 168

MAT Medial Axis Transform. XV, 44, 47–49, 53–57, 60, 126, 128, 131–133, 137, 145, 166, 195

PDMS Product Data Management System. 72

PDP Product Development Process. 1–3, 5–7, 9, 12, 19, 32, 39, 44, 46, 88, 89, 180

PLM Product Lifecycle Management. 6, 16, 21, 32, 72

Introduction

Context of numerical certification of aeronautical structures

Aeronautical companies face increasing needs in simulating the structural behavior of product sub-assemblies. Numerical simulation plays an important role in the Product Development Process (PDP) of a mechanical structure: it allows engineers to numerically simulate the mechanical behavior of this structure submitted to a set of physical constraints (forces, pressures, thermal field, ...). The local or global, linear or nonlinear, static or dynamic analyses of structural phenomena using Finite Element Analysis(es) (FEA) simulations are now widespread in industry. These simulations play an important role to reduce the cost of physical prototyping, to justify and certify structural design choices. As an example, let us consider the last Airbus Program A350. There, a major physical test program is still required to support the development and certification of the aircraft. However, it is based on predictions obtained from Finite Element Models (FEM). Consequently, the test program validates the internal load distributions which have been computed numerically.

Today, FEAs are not restricted anymore to the simulation of components alone; they can be applied to large assemblies of components. Simulation software capabilities associated with optimized mathematical resolution methods can process very large numbers of unknowns derived from their initial mechanical problems. Such simulations require few days of computations, which is an acceptable amount during a product development process in aeronautics. However, it is important to underline that these numerical simulations require the setting of a mathematical model from the physical object or assembly being analyzed. On purpose, the FEA method incorporates simplification hypotheses applied to the geometric models of components or assemblies compared to their real shapes and, finally, produces an approximate solution. To obtain the most faithful results with a minimum bias within a short amount of time, engineers are encouraged to spend a fair amount of time on the generation of simulation models. They have to stay critical with respect to the mathematical method used, the consequences of simplification hypotheses, in order to understand the deviations of the simulation model compared to the real tests to judge the validity of the simulation results.

Some limits faced in structural simulations

Numerical simulations of assemblies remain complex and tedious due to the pre-processing of assembly 3D models available from Digital Mock-Ups(DMUs) that stands as the virtual product reference in industry. This phase is highly time consuming compared to the numerical computations' one. In the past, the use of distinct software tools between the design and simulation phases required generating once again the Computer Aided Design (CAD) geometry of a component in the simulation software. Today, the development and use of DMUs in a PDP, even with a rather small assembly models, bring 3D models at hand for engineers. The concept of DMU was initially developed for design and manufacture purposes as a digital representation of an assembly of mechanical components. Consequently, DMUs are good candidates to support digital analyses of several PDP processes, e.g. part assembly ones. In industry, DMUs are widely used during a PDP regarded as the virtual product geometry reference. This geometric model contains a detailed 3D representation of the whole product structure that is made available for simulation engineers. To prepare large sub-structure models for simulation, such as wings or aircraft fuselage sections; the DMU offers a detailed and precise geometric input model. However, speeding up the simulation model generation strongly relies on the time required to perform the geometric transformations needed to adapt a DMU to FEA requirements.

The pre-processing phase implies reworking all DMU 3D data to collect subsets of components, to remove unnecessary or harmful areas leading to simplified shapes, to generate adequate FE meshes, to add the boundary conditions and material properties as needed for a given simulation goal. All these operations, the way they are currently performed, bring little added value to a PDP. Currently, time and human resources involved in pre-processing CAD models derived from DMUs into FE models can even prevent engineers from setting up structural analyses. Very tedious tasks are required to process the large amount of DMU components and the connections between them, like contact areas.

Commercial software already provide some answers to the interactions between design and behavioral simulation processes for single components. Unfortunately, the operators available are restricted either to interactive geometric transformations, leading to very tedious tasks, or to automated simulation model generation adapted to simple models only. A rather automated generation of complex assembly simulation models still raises real difficulties and it is far too tedious to process groups of components as well as sub-assemblies. As detailed in Chapter 1, these difficulties arise because shape transformations are needed since designers and simulation engineers work with different target models, resulting in the fact that DMUs cannot be easily used to support the preparation of structural analysis models.

Scientific research work has mainly focused on the use of global geometric transformations of standalone CAD components. Few contributions have addressed the au-

tomation of assembly pre-processing (see Chapter 2) leaving engineers to interactively process each assembly component. Aeronautical structures are particularly complex to transform due to the large amount of transformations on hundred thousands of parts and interfaces joints. These operators are still not generic enough to be adapted to engineers' needs, especially when idealizations of components or assemblies must be produced. Indeed, it still is a common practice for engineers to generate interactively their own models because preparation operations are still not generic enough. Consequently, some simulations are not even addressed because their preparation time cannot fit within the schedule of a PDP, i.e. simulation results would be available too late.

Work Purposes

To reach the needs of large assembly simulation model preparation, improvements in processing DMUs are a real challenge in aircraft companies. The contributions in this thesis are mainly oriented toward the transformation of 3D CAD models extracted from a DMU, and their associated semantics, for Finite Element analysis of large assembly structure. To handle large models, it is mandatory that the proposed principles and operators speed up and to automate as much as possible the DMU transformations required. This work will be guided by input DMU data defining the exact content of the simulation models to be built and will use mechanical and geometric criteria for identifying the necessary geometric adaptation.

This research thesis is divided into 6 chapters:

- Chapter 1 describes the current practices in aeronautical industries about the generation, from DMUs, of geometric models supporting the generation of simulation models. It will define the different geometric entities used in CAD software as well as the notion of mechanical analysis using the FE method. Also, it will detail the problematic of DMU geometry preparation for FE assembly models;
- Chapter 2 is a current bibliographical and technological status on the methods and tools proposed for the preparation and adaptation of geometric models for FEA. This analysis will consider the review of component pre-processing as well as their assembly counterpart;
- Chapter 3 presents the proposed contribution to assembly pre-processing based on the recommendations of chapter 1 and the analysis of chapter 2. This approach uses, as input model, an enriched DMU at an assembly level with geometric interfaces between its components, functional properties of these components and, at the component level, a structured volume segmentation using graph structure. From this enriched model, an analysis framework is able to connect the simulation hypotheses with the shape transformations. This chapter will also identify the geometric operators segmenting a component to transform it in accordance with the user's simulation requirements;

- Chapter 4 exposes the principles of the geometric enrichment of a component using a construction graph. An algorithm extracting generative processes from B-Rep shapes will be detailed. It provides a powerful geometric structure containing simple primitives and geometric interfaces between them. This structure contributes to an analysis framework and it remains compatible with an assembly structure containing components and geometric interfaces;

- Chapter 5 details the analysis framework through the exploitation of the construction graph to analyze the morphology of component. Then, geometric operators are specified that can be robustly applied to automate components' and interfaces' shape transformations during an assembly preparation process;

- Chapter 6 extends this approach toward a methodology using the geometric operators previously described that performs idealizations and template based transformations of groups of components. Results of this methodology will also be presented to illustrate it through aeronautical examples that use the transformation operators developed;

Chapter 1

From a Digital Mock Up to Finite Element Assembly Models: Current practices

This chapter presents the problematic of DMU adaptation for the generation of Finite Elements (FE) assembly models. In a first step, the technical context is addressed through the description of the DMU data content. This description deals with the geometrical entities and concepts used to represent 3D CAD components as well as with the representation of assemblies currently available before DMU pre-processing. Then, the notion of mechanical analysis using the FE method is defined and the main categories of geometric models within FEA are described. The analysis of current industrial processes and practical DMU data content highlights the issues regarding assembly simulation model preparation and points out the lack of tools in industrial software to reach the level of abstraction required by FEA, especially when idealizations are needed. The main time consuming shape transformations and the missing information about components' interfaces in DMUs are identified as a starting point to improve the robustness of DMU pre-processing.

1.1 Introduction and definition DMU concept

To speed up a Product Development Process (PDP), as stated in the introduction, aeronautical, automotive and other companies face increasing needs in setting up FE simulations of large sub-structures of their products. Their challenge covers the study of standalone components but it is now expanding to simulate the structural behavior of large assembly structures containing up to thousands of components.

Today, aeronautical companies have to manage a range of products during their entire lifecycle, from their early design phase to their manufacture and even up to their destruction and recycling. The corresponding digital data management concept

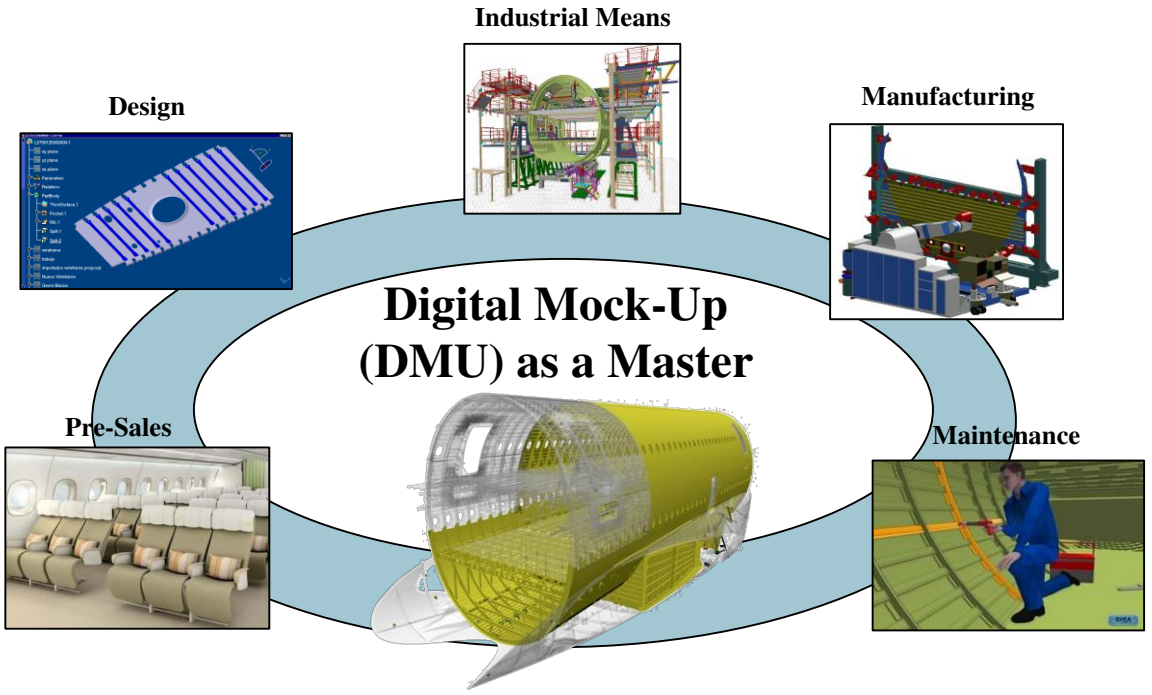


Figure 1.1: The Digital Mock-Up as the reference representation of a product, courtesy of Airbus Group Innovations.

aggregating all the information about each product is called Product Lifecycle Management (PLM). This concept includes the management of a digital product definition for all the functions involved in a PDP. To replace a physical mock up by its digital counterpart, the concept of a virtual representation of a product has been developed, i.e., the Digital Mock Up (DMU) (see Figure 1.1). As Drieux [Dri06] explained, a DMU is an extraction from the PLM of a product at a given time. The DMU was initially created for design and manufacture purposes as a digital representation of an assembly of mechanical components. Consequently, DMUs are convenient to support a virtual analysis of several processes, e.g., part assembly ones. For instance, DMU may be extracted at the manufacturing level, which let engineers to quickly generate and simulate trajectories of industrial robots and to set and validate assembly tolerances. During project reviews of complex products such as an aircraft, DMUs contribute to the technical analysis of a product, as conducted by engineering teams. Connected to virtual reality technology, a DMU can be at the basis of efficient immersive tools to analyze interferences among the various subsystems contained into the corresponding product [Dri06, IML08].

During the design phase, the DMU is considered as the reference geometry of the product representation. It provides engineers all the digital information needed during their PDP. The various CAD models representing different stages of the product during its development or meta data related to specific applications such as manufacturing are

examples of such information. The development and use of DMUs in a PDP bring 3D assembly models at hand for engineers. Because this reference model contains detailed 3D geometry, it offers new perspectives for analysts to process more complex shapes while speeding up their simulation model generation up to the FE mesh. However, speeding up the simulation model generation strongly relies on the time required to perform the geometric transformations needed to adapt the DMU to FE requirements.

In order to understand the challenges involved in the preparation phase from DMUs of large sub-structure models for FE simulations, it seems appropriate to initially present the various concepts and definitions related to the Finite Element Analysis (FEA) of mechanical structures as well as those related to the current models and information available in DMUs. This chapter describes the current practices regarding shape transformations needed to generate the specific geometric models required for FEA from a DMUs model. Starting from the theoretical formulation of a mechanical analysis to the effective shape transformations faced by engineers to generate FE models, this chapter raises the time consuming preparation process of large assembly structures as an issue. This is detailed at section 1.5 and refers to the identification of key information content during current FE simulation preparation processes from two perspectives: a component point of view as well as an assembly point of view. In the last section 1.7, the research objectives are presented to give the reader an overview of the research topic addressed in this thesis.

1.2 Geometric representation and modeling of 3D components

A DMU is straightforwardly related to the representation of the 3D components contained in the product. As a starting point, this section outlines the principles of mathematical and computer modeling of 3D solids used in CAD software. Also, it describes the common schemes available for designers to generate components through a construction process. Because a component is used in a DMU as a volume object, this section focuses on solids' representations.

1.2.1 Categories of geometric families

Prior to explanations about the common concepts for representing 3D components, it is important to recall that the geometric model describing a simulation model contains different categories of geometric entities used in the CAD and FEA software environments. These entities can be classified, from a mathematical point of view, in accordance to their manifold dimension.

0-dimensional manifold: Point

These entities, the simplest geometric representation, are not intended to represent the detailed geometry of components. However, they are often used in structural analysis, as abstraction of a component, i.e., its center of mass or a key point of a component where concentrated forces are applied, ... They are also frequently encountered to represent interfaces between aeronautical systems and structures in a DMU. They are also the lowest level entity in the description of a component's solid model.

1-dimensional manifold: Line, Circle, Curve

These entities, such as lines, circles and more generally curves, are mainly involved in the definition of models of higher dimension like surfaces. In structural analysis, they represent long and slender shapes, e.g., components behaving like beams, with the complement of section inertia. During a solid modeling process, they are part of the definition of 2D sketches (see Figure 1.5 for an example of sketch based form feature), or as profile curves in other shape design processes. Also, they represent the location of geometric interfaces between components, e.g., the contact of a cylinder onto a plane.

2-dimensional manifold: Surface

Surfaces are used to represent the skin, or boundary, of a 3D object. Initially, they were introduced to represent complex shapes of an object, commonly designated as free-form surfaces. Polynomial surfaces like Bézier, B-Spline, NURBS (Non-Rational B-Spline), Coons surfaces are commonly used for modeling objects with curved surfaces and for the creation of simulation models, e.g., CFD simulations for aerodynamics or simulations using the isogeometric paradigm. Here, surface models will be essentially reduced to canonical surfaces, i.e., plane, sphere, cylinder, cone, torus, which are also described by classical implicit functions. This restriction is set for simplicity purposes though it is not too restrictive for mechanical components because they are heavily used. In structural analysis, using a surface model is frequent practice to represent idealized models equivalent a volume component resembling a sheet. The notion of idealization will be specified in subsection 1.3.2. Even if a surface model can represent complex shapes, they are not sufficient to represent a 3D object as a volume, which requires an explicit representation of the notion of inside, or outside.

3-dimensional manifold: Solid

A solid contains all the information to define comprehensively the volume of the 3D object it represents. Based on Requicha's mathematical definition [Req77, Req80], a solid is a subset of the 3D Euclidian space. Its principal properties are:

- A solid have a homogeneous three dimensionality. It contains a homogeneous interior. Solid's boundary cannot have isolated portions;

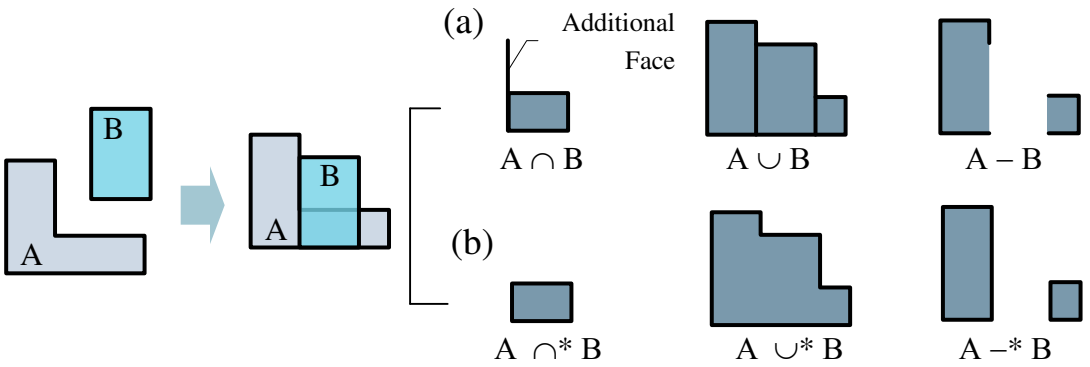


Figure 1.2: Boolean operator of two solids: (a) Conventional boolean operator produce additional face, unclosed boundary. (b) CAD uses regularized boolean operator producing valid solid

- A solid must be closed. When applied to solid, rigid motions (translation, rotation) or operations that add or remove material, must produce others solids;
- The boundary of a solid must determine unambiguously the interior and exterior of the solid.

To describe a solid, topological properties are mandatory in addition to the geometric entities defining this object. This requirement is particularly important when describing complex shapes because they are generated using a process that combines simple primitives to progressively increase the shape complexity until reaching the desired solid. Indeed, the principle of generation process exists for complex free-form surfaces is similar to that of complex solids. During the design process, the constructive processes used to combine elementary primitives are key information to enable efficient modification processes that are frequently required during a PDPIt is commonly admitted that 80% of the design time is spent on modifications processes.

1.2.2 Digital representation of solids in CAD

Geometric modeling is the core activity of a CAD system. It contains all the geometric information describing 3D objects. Although there are various digital representations of 3D components falling into the category of solids, the two major representations used in CAD software are detailed in the following paragraphs.

Constructive Solid Geometry (CSG) representation

This representation designates Constructive Solid Geometry approaches devoted to the design and generation of 3D components. It is important to note that the 'usual' set of Boolean operations cannot be directly applied on solid model. Otherwise, it would create invalid solids. As illustrated on Figure 1.2a, the conventional intersection

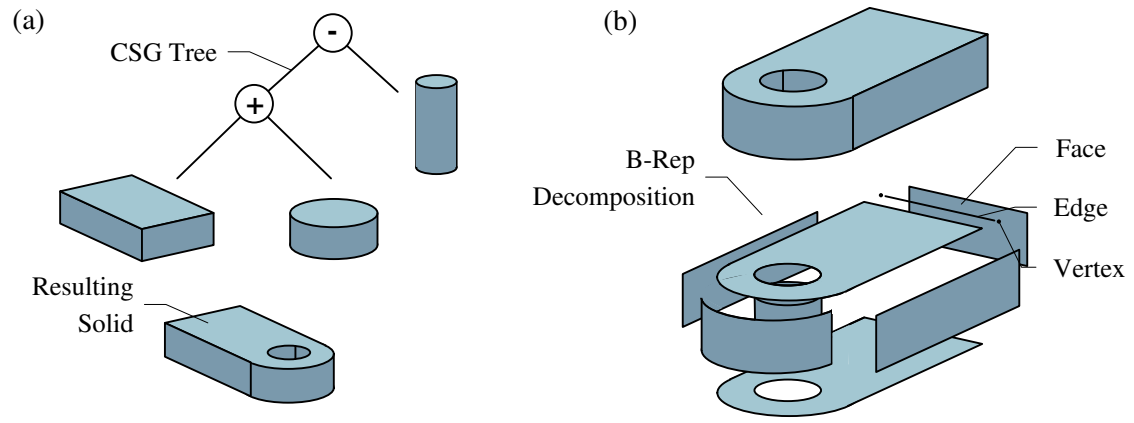


Figure 1.3: (a) Representation of the construction tree of a CSG component. (b) B-Rep solid decomposed into faces, edges and vertices (after Stroud [Str10]).

operator applied to two solids would create a non regular solid with an isolated face, in Figure 1.2b the intersection of two cubes would generate a face and not a solid or the empty set. Therefore, it is necessary to define a new set of Boolean operations, the so-called regularized set intersection, union and difference (see Figure 1.2b). These operators are a modified version of the conventional operators and will be used in the algorithm presented in chapters 4 and 5.

The CSG approaches represents a solid as a sequence of Boolean operations of elementary solids (cylinder, sphere, extrusion, revolution), i.e., primitives (see Figure 1.3a). The modeler stores primitives (cylinders, cubes, ...) and operations that have been applied to them, essentially regularized Boolean operations (union, intersection, difference). It can be visually represented as a tree structure but a CSG representation does not necessarily form a binary tree. The advantage of this model is to give a structure which can be easily modified, if the modification is compatible with the construction tree. The location of a simple primitive, e.g., a hole created with the subtraction of a cylinder, can be easily changed without modification of the CSG structure. Weaknesses are: the difficulty to represent complex geometric shapes with free-form surfaces, the complete tree re-evaluation under modifications and the non uniqueness of this tree with regard to a given shape.

Boundary representation (B-Rep)

In a B-Rep representation, the CAD kernel processes the skin of the object and the inside/outside of material. The B-Rep model contains the result of the operations, i.e., the information defining the shape of the solid (see Figure 1.3b). The volume of a solid is represented by a set of surfaces describing its boundary. Two categories of information are stored in a B-Rep model, a topological structure and a set of geometric entities:

- The geometric information: it consists into a set of surfaces defining the boundary of the solid and locating it in 3D space. These surfaces are bounded by trimming curves;

- The topological information: this datastructure enables the expression of the mandatory topological properties, i.e., closure, orientation, that leads to the description of shells, faces, wires, edges, vertices, expressing the adjacency relationships between the topological entities. It can be represented using incidence graphs such as face-edge and edge-vertex graphs.

In a B-Rep model, the set of surfaces is closed and Euler operators express the necessary conditions to preserve the consistency of a solid's topology during modifications. The advantage of this representation holds in its ability to use non-canonical surfaces, i.e., NURBS, allowing a user to represent more complex shapes than CSG representation. Among the disadvantages of B-Rep models, the representation of the solid's boundary contains information only about its final shape and this boundary is not unique for given shape. Today, the B-Rep representation is widespread in most of CAD geometric modelers and it is associated with a history tree to enable the description and use of parametric models.

Nowadays, CAD modelers incorporate the B-Rep representation as well as Boolean operators and the B-Rep representation is the main representation used in aeronautical DMUs. In this thesis, the input CAD model of a 3D component is considered as extracted from a DMU and defined as a solid via a B-Rep representation.

Representation of manifold and non-manifold geometric models

To understand the different properties used in CAD volume modelers and Computer Aided Engineering (CAE) modelers, the notions of manifold solid and non-manifold objects have to be defined. One of the basic properties of a CAD modeler representing solids is that the geometric models of 3D components have to be two-manifold to define solids. The condition for an object to be two-manifold is that, 'at every point of its boundary, an arbitrary small sphere cuts the object's boundary in a figure homomorphic to a disc'. This condition ensures that a solid encloses a bounded partition of the 3D space and represents a physical object. Practically, in a B-Rep representation, the previous condition reduces partly to the another condition at every edge of a manifold solid where it must be adjacent to two faces exactly. Because a solid is a two-manifold object, its B-Rep model always satisfies the Euler-Poincaré formula as well as all the associated operators performing the solid's boundary transformations required during a modeling process:

$$v - e + f - h = 2 * (s - g) \tag{1.1}$$

where v, e, f, h, s and g represent the numbers of vertices, edges, faces, hole-loops and the numbers of connected components (shells) and the genus of the solid, respectively.

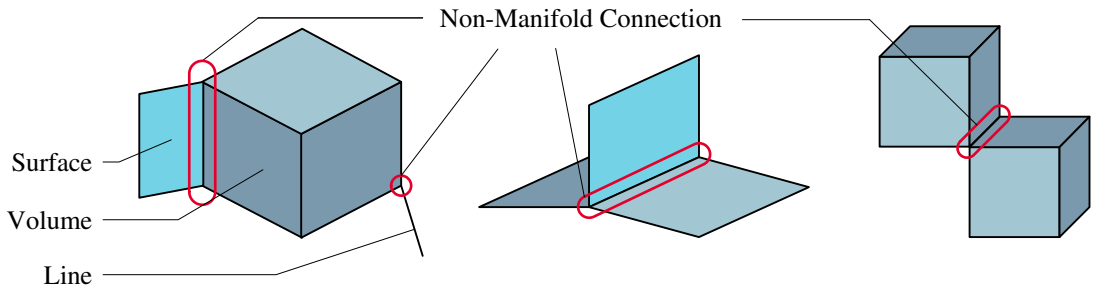


Figure 1.4: Examples of non-manifold geometric models.

As a difference, an object is said to be non-manifold when it does not satisfy the conditions to be a manifold. To address the various needs of object representation, the concept of manifold has been extended to represent a wider range of shapes as needed through a PDP. As illustrated in Figure 1.4, a non-manifold geometric modeling kernel incorporates the ability to describe geometric regions of different manifold dimensions, connected or not along other geometric regions of lower manifold dimensions. Consequently, an edge can be adjacent to more than two faces. However, some basic properties of solids are no longer valid, which increases the difficulty in defining the consistency of such models. In the context of the Computer Aided Design to Finite Element Analysis(es) (CAD-FEA), this also referred to as 'cellular modeling' [TNRA14] and few geometric modeling kernels, natively incorporate this category of models [CAS14] These models are commonly used in structural analysis where surfaces often intersect along more than one edge (see Figure 1.11c). Therefore, CAE software proposes datastructures to generate non-manifold geometry. However, most of the commercial FEA softwares contains manifold geometric modelers with extensions to be able to model non-manifold objects, which does not bring the desired end-user performances. Here, the input CAD models are considered as manifold solids and the generated models for FEA can be non-manifold.

1.2.3 Complementary CAD software capabilities: Feature-based and parametric modeling

CAD software incorporate a volume geometric modeling kernel to create manifold solids and a surface modeler to enable the generation of two-manifold objects with free-form surfaces, i.e., two-manifold with boundary objects. CAD tools are essential for the generation of DMU because they are used to design and to virtually represent the components of a product. In addition to the presentation of the geometric models used in CAD software, (see Section 1.2.2), it is also crucial to mention some CAD practices contributing to the design of mechanical components. This will help understanding the additional information associated to B-Rep models which are also available in a DMU.

Concept of feature:

As stated in Section 1.2.2, B-Rep models only describe the final shape of solids. To ease the generation process of 3D mechanical models and to generate a modeling history, CAD software uses pre-defined form features as primary geometric regions of the object [Sha95]. A feature is a generic concept that contains shape and parametric information about a geometric region of a solid. The features can represent machining operations such as holes, pockets, protrusions or more generic areas contributing to the design process of 3D components like extrusions or revolutions.

Generative processes:

The following chapters of this thesis use the term "generative processes" to represent an ordered sequence of processes emphasizing the shape evolution of the B-Rep representation of a CAD component. Each generative process corresponds to the generation of a set of volume primitives to be added or to be removed from a 3D solid representing the object at one step of its construction.

Features Taxonomy: The features of solid modeling processes can be categorized into two sets [Fou07]:

- Features independent from any application:
  - Geometric entities: points, axes, curves, sketches;
  - Adding/removing material: extrusion, revolution, sweeping. Boolean operators;
  - Surface operations: fillets, chamfers, fillings;
  - Repetitions, symmetries.
- Features related to an application (hole drilling, sheetmetal forming, welding).

As an example, a material addition extrusion feature is illustrated in Figure 1.5. Its generative process consists in drawing a 2D sketch using lines, circles or planar curves to define a planar face and then, to submit it to a translation using an extrusion vector to generate a volume.

The principle of feature-based modeling is to construct a part "from a simple shape to a complex one" and it is similar to the CSG principle. As illustrated in Figure 1.5 and, more generally, in Appendix A where a complete part design is represented, the user starts with the design of simple volumes to represent the overall solid's shape of a component and to add progressively shape details such as fillets and holes to reach its final shape. This qualitative morphological approach to a solid's construction process

can be partly prescribed by company modeling rules but a user stays particularly free to choose the features and their sequence during a construction process. This is a consequence of a design process enabling the user to monitor step by step, with simple features, the construction process of a solid.

In CAD software, the sequence of features to create a component is represented and stored in a construction tree (see Figure 1.5).

Dependences between features and construction tree:

The construction tree of a solid is connected to the notion of parametric modeling. In addition to the feature concept, parametric modeling has been introduced in CAD software to enable the regeneration of a solid when the user wants to apply a local shape modification. With parametric modeling, the user input defines the geometric constraints and dimensions of a feature in relation with others. In most cases, the localization of a new feature is not applied in the global coordinates system of the solid. A feature uses an existing geometric entity of the solid, e.g., a planar face, as basis for a new sketch (see Figure 1.5). The sketching face creates another dependency between features in addition to the parent/ child relationships that are stored in the construction tree.

Conclusion

Solid modeling is a way to represent a digital geometric model of a 3D component. The B-Rep representation used in commercial CAD software allows a user to design complex shapes but it provides only a low level volume description because it does not give a morphological description of a solid. As a complement, feature modeling is easy to learn for a CAD user because it allows him, resp.her, to naturally design mechanical components without an in-depth understanding of CAD modeling kernel. Construction trees structure a 3D object using simple feature models, extending the use of B-Rep models with a history representing their construction processes. Parametric modeling is also efficient to produce a parameterized representation of a solid to enable easy modifications of some of its dimensions.

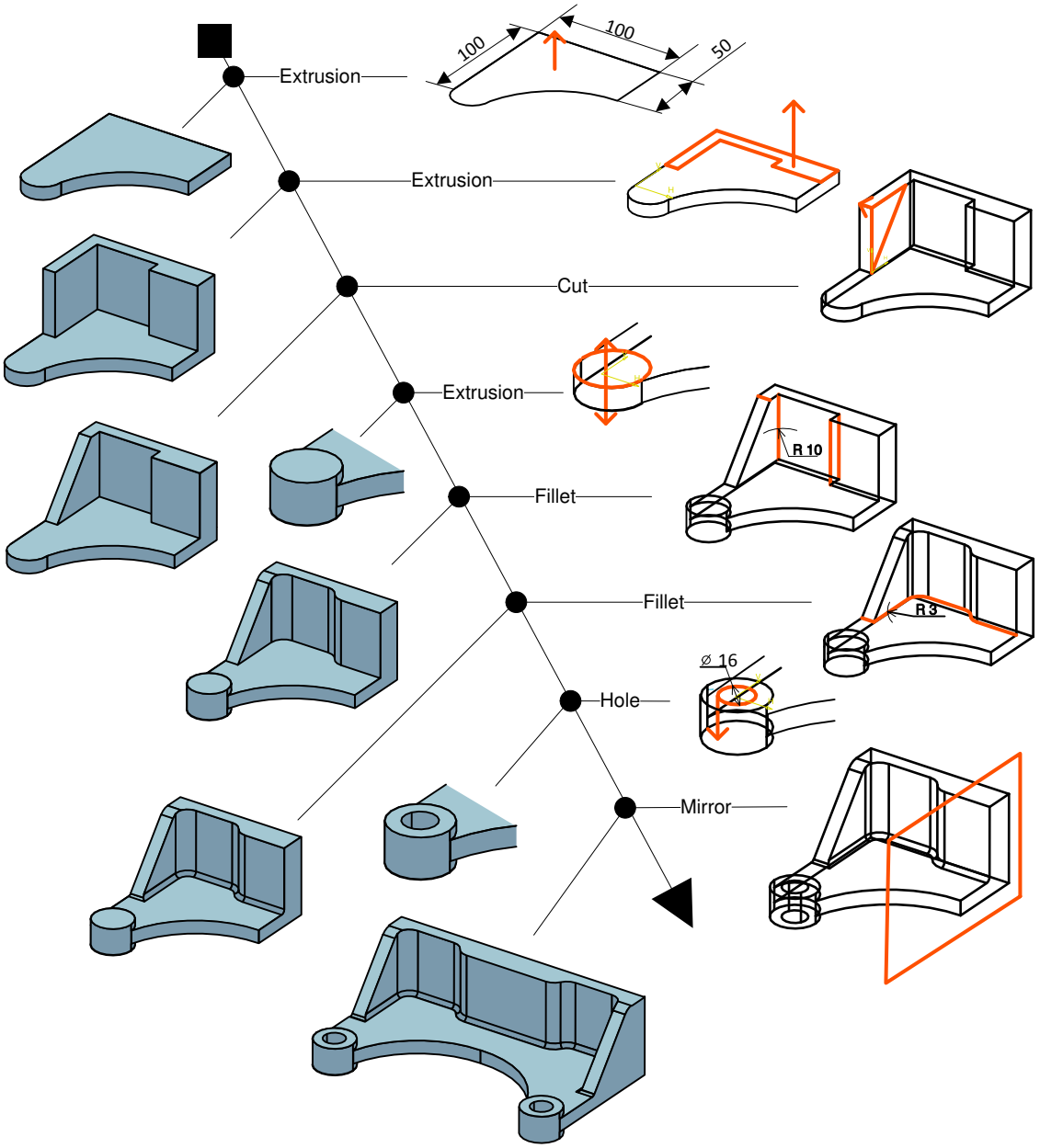


Figure 1.5: CAD construction process using form features. The modeling sequence is stored in a construction tree.

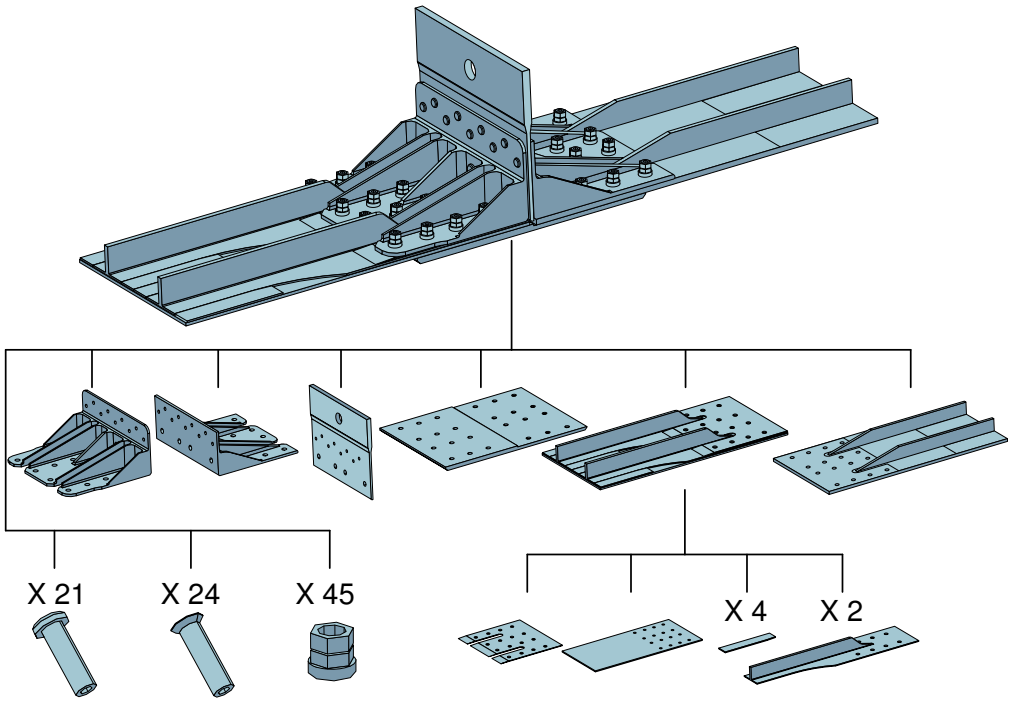


Figure 1.6: Example of an aeronautical CAD assembly: Root joint model (courtesy of Airbus Group Innovations).

1.3 Representation and modeling of an assembly in a DMU

Any mechanical system is composed of different components assembled together with mechanical joints. This section aims at presenting how an assembly is represented and created in a DMU. It underlines how an assembly is processed with a non-formal conventional representation. In particular, this section deals with the current content of a DMU extracted from a PLM in an aeronautic industrial context (data that are actually available for structural simulation).

1.3.1 Effective DMU content in aeronautical industry

Assembly structure in a CAD environment

In a CAD software, an assembly is a structure that organizes CAD components into groups. Each component contains the geometrical and topological data as described in Section 1.2.2. Then, the component is instantiated in the assembly structure as many times as it should appear. The Figure 1.6 represents an aeronautical structure (wing fuselage junction of an aircraft) with a sub-assembly for each composite part and instantiation of standard components such as screws and nuts.

To create an assembly structure in 3D, the user iteratively positions components in 3D space relatively to other components, (axis alignment of holes, surface mating, ...). These connections between components, called position constraints, connect degrees of freedom from each component involved in the corresponding constraints. However, these constraints may not represent the common geometric areas connecting the corresponding components. For instance, to connect a screw with the through hole of a plate, a coaxiality constraint can be applied between the cylindrical surface axis of the screw and the hole axis, independently from any contact between the surfaces. The radii of the cylindrical surfaces of these two components are not involved, i.e., any screw can be inserted in the hole. This does not match the reality and can lead to assembly inconsistencies.

In a CAD environment, the assembly structure is stored in a product tree which connects each of its components to others with assembly constraints. On top of the B-Rep representation, each component can contain complementary information such as a name or a product reference, a contextual description of the component's function, modification monitoring information, color, material designation. During a product design process, a CAD environment also offers capabilities to share external references to other components not directly stored in a product structure, to parameterize components' dimensions with mathematical formulas, ...

DMU evolution during a PDP

All along the product construction process in a PDP, its digital representation evolves. The information related to the product definition such as its 3D geometric representation gets modified. In addition, the product development process is shared among several design departments that address different engineering areas: the product mechanical structure, electrical systems, ... All these areas have to share their design information and sub-assemblies definitions. Whether it is geometry or parameters, this information should be integrated in a common product representation. The DMU concept is a means to support the geometric definition and evolution of components while maintaining the product assembly updated. As an example, depending on the maturity of the product, a DMU can be reduced to a simple master geometry at the early stage of the product design using functional surfaces or, it can contain the full 3D representation of all its components as required for manufacturing.

All these needs require that a Product Data Management System (PDMS) be used to support the definition and evolution of a DMU. The PDMS structures the successive versions, adaptions to customer requirements as well as the various technical solutions that can be collectively designated as variants. A product is represented as a tree referencing the CAD components and their variants. The various product sub-assemblies of the design team, initially created in a CAD environment, are transferred and centralized in the PDMS. It is based on this environment that an engineer can formulate a request to extract a DMU used as input model for his, resp. her, simulations.

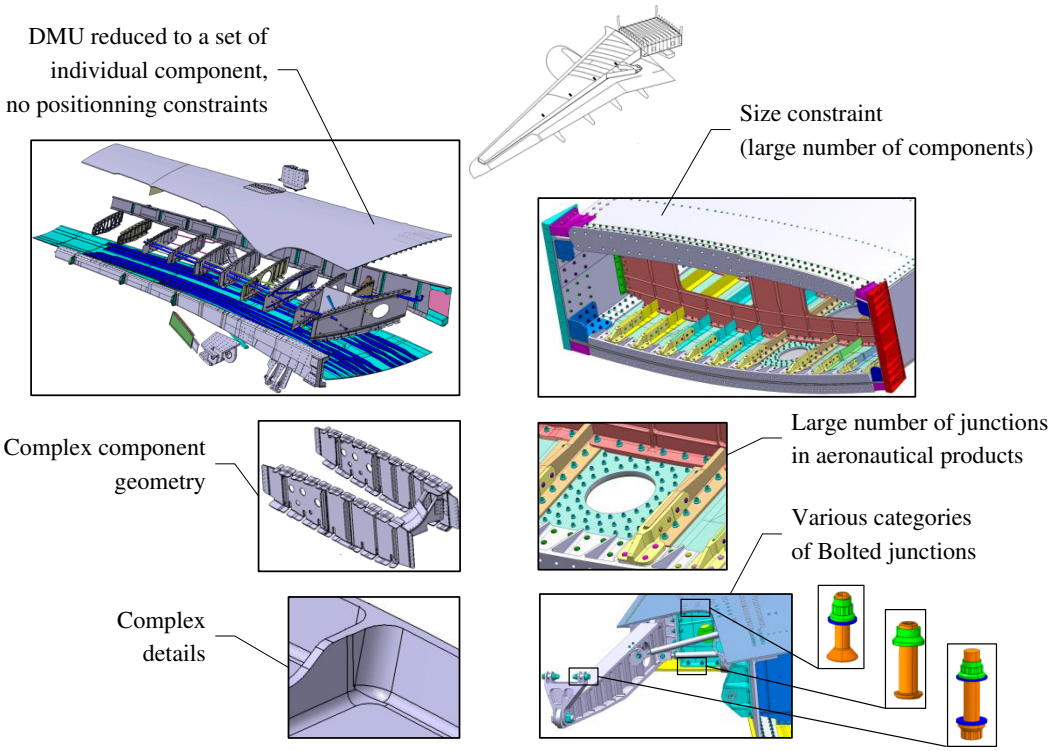


Figure 1.7: Example of complex DMU assembly from Alcas project [ALC08] and Locomachs project [LOC16].

The DMU content and management in the aeronautical industry: a pragmatic point of view.

Today, as described previously, a DMU stands for the reference geometric representation of a product used by structural and systems engineers. A DMU is the input model to generate their simulation models. In practice however, the information of a DMU extracted from the PDMS reduces to a set of CAD components positioned in 3D space with respect to a global reference frame and a tree structure representing a logical structure of this product [Fou07, FL*10]. This fair loss of information originates from:

- The size and the fragmented location of a DMU: it contains a large amount of components (see Figure 1.7) created by different design teams during a PDP, e.g., in aeronautics, the extraction of a DMU from the PDMS requires one day (no centralized data);
- The robustness of a DMU: Positioning constraints between components are not available. Components are standalone objects in a common reference frame. A DMU is an extraction from the PDMS at a given time. During the evolution of a PDP, engineers cannot maintain the interfaces between the geometric models

of the components; the corresponding geometric constraints, if they were set, have to be removed because their management becomes too complex. As an example, if a component is removed, the removal of its corresponding geometric constraints could propagate other modifications throughout the whole assembly. Additionally, the amount of geometric constraints gets very large for complex products and their consistency is still an open problem [LSJS13]. It can reach more than three hundreds for an assembly with less than fifty components. This is what motivates the solution to locate all components into a global coordinate system. Consequently, each component is positioned independently of the others, which increases the robustness of the DMU regarding to its modifications even though the consistency of the DMU becomes more difficult to preserve.

As a result, if a product is complex and has a large amount of components created by different design teams during a PDP, the PDMS does not contain information specifying assemblies. Information about proximity between components is not available. The relational information between parts in an assembly created in a CAD environment, e.g., the assembly constraints, is lost during the transfer between the CAD environment and the PDMS. The DMU is restricted to a set of individual CAD components and a tree decomposition of a product. As Drieux explained in its DMU analysis [Dri06], a DMU is a geometric answer to design specifications, it does not contain multiple representations adapted to the various users' specifications during a PDP, including the structural analysis requirements.

1.3.2 Conventional representation of interfaces in a DMU

In order to carry on the pragmatic analysis of a configuration where a simulation engineer receives a DMU as input model, this section defines the notion of assembly interfaces between components and their conventional representations in industry.

Lacking conventional representation of components

Shahwan et al. showed [SLF*13] that the shapes of digital components in a DMU may differ from the shape of the physical object they represent. These differences originates from a compromise between the real object shape that can be tedious to model and the need for their shape simplifications to ease the generation of a DMU. This is particularly true for standard parts such as components used in junctions (bolted, riveted). Regarding their large number, each geometric detail, e.g., a threaded area, are not represented because it would unnecessarily complicate the DMU without improving its efficiency during a design process. Since there is no standard geometric model of assembly components used in 3D junctions, each company is likely to set its own representation.

As illustrated in Figure 1.8, in large aeronautical DMUs, bolted junctions as well as

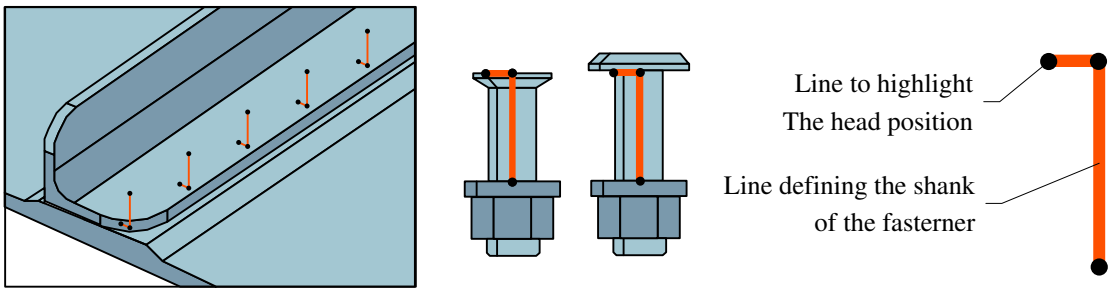


Figure 1.8: Representation of a bolted junction in a structural DMU of an aircraft.

riveted junctions may be represented with a simplified representation defined with two perpendicular lines. This representation is sufficient to generate an equivalent volume model using basic information of bolt type, nominal diameter and length. However, no information exists about the connections between the bolt and the junction's components the bolt is related to. There is neither a logical link between the screw and nut with the plates they connect nor a geometric model of the interface between the screw and nut forming the junction. More generally, the lack of geometric interface exists for every interface between assembly components. As explained at Section 1.5.3, this poor representation complicates the generation of equivalent simulation models. This complexity issue applies also to deformable components[1], which are represented under an operating configuration.

Definition of interfaces between components

Based on the description of the DMU content given at Section 1.3.1, there is no explicit information about the geometric interaction between components. Even the content of geometric interfaces between components can differ from one company to another. In this thesis, the definition of Conventional Interface (CI) of Léon et al. [FL*10, LST*12, SLF*13] is used. From their DMU analysis, they formalized the representation of conventional interfaces between components. To cover all the possible interactions between two B-Rep objects C_1 and C_2, they classified the CIs into three categories (see Figure 1.9):

1. *Contacts:* these are configurations where the boundary surfaces of the components C_1 and C_2 and relatives positions of these components are such that: $\partial C_1 \cap \partial C_2 = S \neq \varnothing$ and $\partial C_1 \cap^* \partial C_2 = \varnothing$ where ∂C_1 and ∂C_2 represent the boundary surfaces of C_1 and C_2, respectively. S refers to one or more geometric

[1]Here, deformable components refer to a category of components with plastic or rubber parts whose displacements under loading conditions are of a magnitude such that the designer take them into account when setting up the DMU. This is to oppose to metal parts where their displacements are neglected.

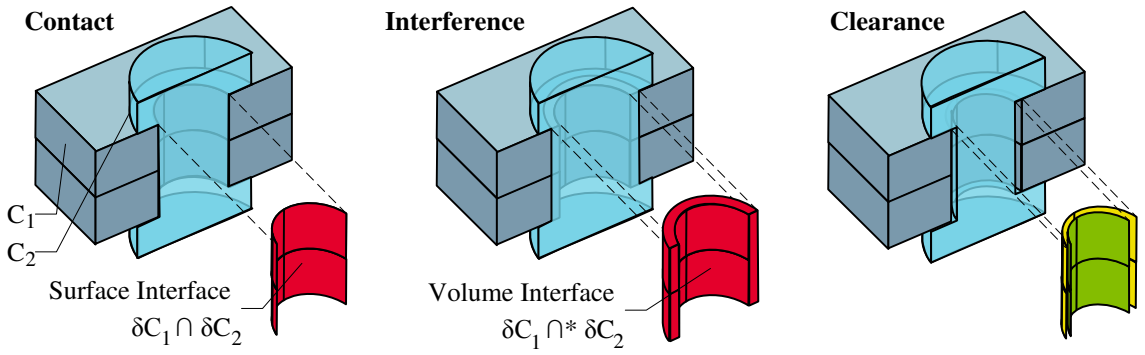


Figure 1.9: Classification of Conventional Interfaces (CI) under contact, interference and clearance categories.

elements that can be surface-type, line-type or point-type. Figure 1.9a illustrates contacts between CAD components;

2. *Interferences:* these are configurations where the boundary surfaces of the components C_1 and C_2 and the relative positions of these components are such that: $\partial C_1 \cap^* \partial C_2 = C_{12} \neq \varnothing$ where C_{12} is the intersection volume. Interferences are detected and analyzed during a DMU review to ensure that there is no physical integration problem between components. However, according to Léon et al., interferences, also named clashes, may occur when components' shapes are simplified with respect to their physical models (see Figure 1.9b), or in case of incorrect relative positions of components. Interferences resulting from these partial positions make a DMU virtually inconsistent, which requires user's analysis. Interferences between standard components generate specific classes of interferences, which is used to process DMUs in the present work;

3. *Clearances:* they represent 3D domains without a clear geometric definition, which is difficult to identify and to represent, (see Figure 1.9c). In this work, clearances are considered as functional clearances and are identified as design features.

The concept of CI can be used in our assembly context, since it is independent from any modeling context. Section 3.3 explains how CI can be extracted from a DMU.

Conclusion

The development and use of DMUs in a PDP bring 3D models at hand for engineers. The DMU extracted from a PLM contains the complete geometric representation of the product using a B-Rep representation and CAD construction trees. Complex shapes are directly available without having to be rebuilt them in a simulation software environment. However, due to considerations of robustness and size, a DMU is reduced

to a set of isolated components without an explicit geometric representation of the interfaces between them.

1.4 Finite Element Analysis of mechanical structures

This section aims at introducing some principles of the Finite Element Method (FEM) for structural analysis. Because the scope of this thesis covers the pre-processing of geometrical data for FEA, this section does not detail the resolution method but focuses on the input data required by FEA. First of all, it introduces the concept of mechanical model for FEA. Then, it enumerates the data needed for FEA ranging from the geometric model of each component using a FE mesh to the representation of connections between meshes as required to propagate displacements and stress fields over the assembly that stand for the mechanical models of the interfaces between components. Subsequently, it describes the industrial approach to various mechanical analyses of an aircraft structure at different levels of physical phenomena from large coarse models representing global deformations to detailed small assemblies devoted to the analysis of stress distributions, as examples. Within each category, the geometric models representing the components and their connections are described.

1.4.1 Formulation of a mechanical analysis

The goal of a numerical simulation of the mechanical behavior of a structure is to anticipate or even supersede a physical test. It allows engineers to simulate a mechanical behavior of a virtual structure, i.e., without the existence of the real structure.

The mechanical analysis process

Independently from a resolution method, e.g., the finite element analysis or the finite difference method, as stated in Fine [Fin01], the mechanical analysis process may be split into three main steps (see Figure 1.10):

1. The formulation of the model behavior:
   Just as in a physical test, each virtual simulation has a specific objective: a simulation objective (type of behavior to be observed such as displacements in a particular area, maximal loads under a prescribed mechanical behavior, accuracy of the expected results, ...). As Szabo [Sza96] describes, the first formulation phase consists in building a theoretical model integrating the mechanical behavior laws representative of the physical system. The analyst specifies and identifies the key attributes of the physical system and the characteristic values of the mechanical behavior: the simulation hypotheses. Then, the analyst applies a

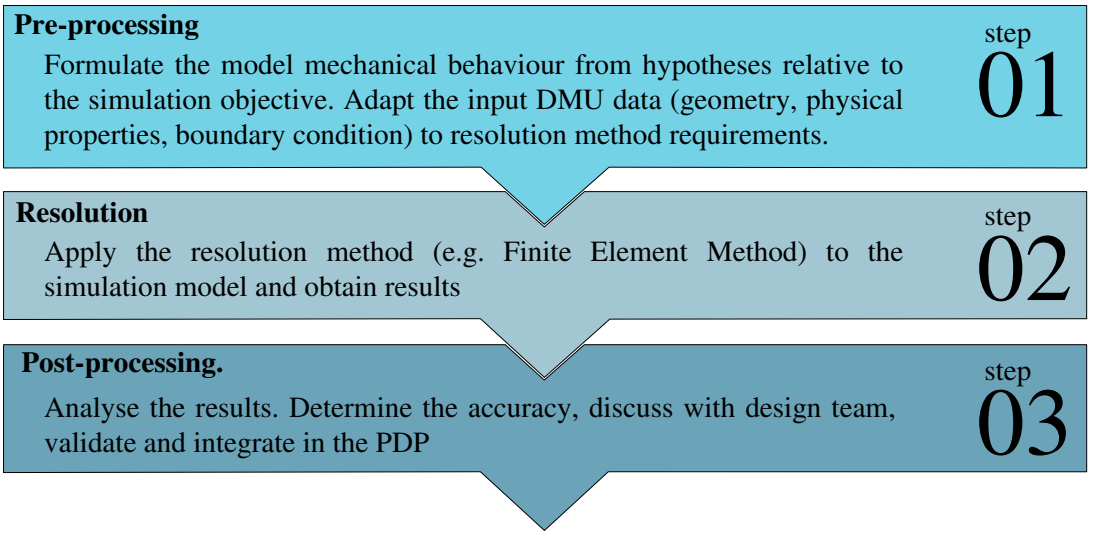


Figure 1.10: Process flow of a mechanical analysis.

set of modeling rules related to the simulation hypotheses in order to create a reduced numerical simulation model ready to be sent to the resolution system. The choice of the modeling rules implies decisions on the mechanical behavior of the structure. When defining the shape of the structure derived from its real shape (see Section 1.4.2) and setting up the constraints and hypotheses related to analytical resolution methods, the mechanical engineer limits its range of observations to the simulation objectives. In practice, the formulation of the model behavior may be viewed as the transformation of the DMU input, which is regarded as the digital representation of the physical structure, into the numerical simulation model. Section 1.5 gives details of this crucial integration phase;

2. The resolution of the model behavior:
   Once the simulation model is generated, the mechanical engineer launches the resolution process. This phase is performed automatically by the CAE software. Currently, the main resolution method used for structural analysis is the FEM, which sets specific constraints at the level of the mesh generation process;

3. The results analysis:
   Once the resolution process has ended, the mechanical engineer has to analyze the results, i.e., the solutions fields computed and the output parameters that can be derived from these fields. He, resp. she, determines the solutions's accuracy, discusses with design teams to decide about shape modifications, validates and integrates the results in the PDP.

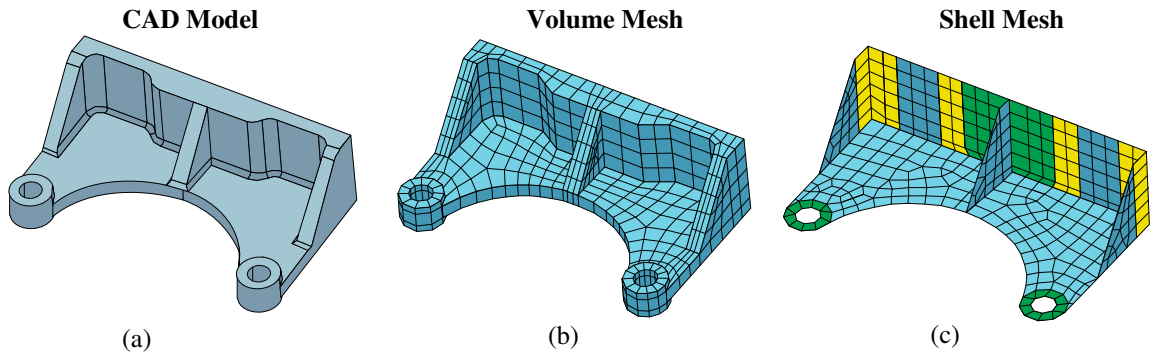


Figure 1.11: Example of FE mesh models: (a) CAD initial model of a structure, (b) Meshed model with 3D volume elements,(c) Meshed model with idealized 2D shell elements.

1.4.2 The required input data for the FEA of a component

Although other resolution methods exist (analytical or numerical, e.g., finite difference method), in mechanical simulation, the FEM is a method widespread in industry. The FEM is a general numerical method dedicated to the resolution of partial differential equations and its applicability is not restricted to structural simulation, it covers thermal, electromagnetism, thermodynamics, ... Many documents exist which relate in detail the principles of this method, reference books of Zienkiewicz [ZT00], Bathe [Bat96] are among them. This section concentrates on the data requirements of the method to formulate a simulation model and addresses pragmatically how the engineer can collect/generate these data.

Geometry: Finite Element Mesh

To solve partial differential equations applied to a continuum, i.e., a continuous medium, the FEM defines an equivalent integral formulation on a discretized domain. This discrete domain is called a Finite Element Mesh and is produced by decomposing the CAD model representing the structure into geometric elements of simple and well known geometry, i.e., triangles, tetrahedra forming the finite elements, ..., whose individual physical behavior reduces to a simple model (see Figure 1.11). When the structure is subjected to a set of physical constraints, the equilibrium equations of the overall structure percolate through all the elements once they have been assembled under a matrix form.

The model size, i.e., the number of finite elements, has a direct influence on the computation time to obtain the solution fields and it may introduce approximation errors if not set correctly. The engineer must efficiently identify the right level of mesh refinement related to the mechanical phenomenon he, resp. she, wants to observe. In practice, to ease the meshing phase, the input CAD geometry is simplified to adapt its shape to the simulation objectives and the mesh generation requirements. If this

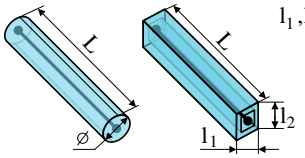
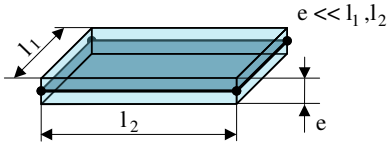
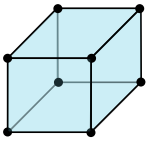
Finite Element	Geometry	Morphological properties
1D-element: Beam		Long and slender sub domain having two dimensions that are small enough compared to the third one. These two dimensions define the beam section parameters.
2D-element: Shell, plate, membrane		Thin sub domain having one dimension which is small compared to the two others. This dimension defines the thickness parameter.
3D-element: Volume		Sub domain without any specific morphological property that must be processed with a three-dimensional mechanical behavior.

Table 1.1: Categories of Finite Elements for structural analyses.

simplification is carried out properly, it would not only generate a good mesh but this mesh is obtained quickly also. This simplification phase incorporates shape transformations and all their inherent issues are discussed in Section 1.5.

Finite Element Choice and families

When setting up a simulation model, the choice of finite elements is essential. Each finite element has an approximation function (polynomial function) which has to locally approximate at best the desired solution. As explained in Section 1.4.1, it is the engineer who chooses the type of finite element in adequacy with the prescribed simulation objectives. There are many types of finite elements to suit various applications and their selection is conducted during the early phase of the CAD model pre-processing. It is a matter of compromise between the geometry of the components, the desired accuracy of the simulation results as well as the computation time required to reach this accuracy.

Figure 1.1 presents the main categories of finite elements classified in accordance with their manifold properties (see Section 1.2.1).

Idealized elements

Based on the shell theory of Timoschenko [TWKW59], specific finite elements are available in CAE software to represent a thin volume, e.g., shell elements. These elements can significantly reduce the number of unknowns in FE models, leading to

a shorter computation time compared to volume models. Also, using shell elements rather than volume ones gives access to different mechanical parameters: section rotation and stress distribution in the thickness is implicitly described in the element. Rather than discretizing a volume into small volume elements, it is represented by its medial surface (see Table 1.1 2D-element). The thickness becomes a numerical parameter associated with the element. Long and slender sub domains can be processed analogously. A beam element is well suited to represent these volume sub domains using an equivalent medial line (see Table 1.1 1D-element). From the sections of these volumes, their inertia parameters are extracted and they become numerical parameters assigned to the beam elements.

Such elements imply a dimensional reduction of the initial volume, a 1-dimensional reduction for shells and 2-dimensional reduction for beams. This modeling hypothesis is called idealization. In a CAD-CAE context, the idealization refers to the geometric transformation converting a initial CAD solid into an equivalent medial surface or medial line which handles the mechanical behavior of a plate, a shell or a beam, respectively. This geometrically transformed model is called idealized model. Such an example is given on Figure 1.11c. Idealized sub domains are particularly suited to aeronautical structures, which contain lots of long and thin components (panels, stringers, ...). Using idealized representations of these components can even become mandatory to enable large assembly simulations because software license upper bounds (in terms of number of unknowns) are exceeded when these components are not idealized. However, Section 1.5.3 illustrates that the practical application of such an idealization process is not straightforward. The sub domains candidates for idealization are subjected to physical hypotheses:

- The simulation objectives must be compatible with the observed displacements or stress field distributions over the entire idealized sub-domains, i.e., there is no simulation objective related to a local phenomenon taking place in the thickness or section of an idealized domain;
- The sub domains satisfy the morphological constraints of idealization hypotheses, e.g., a component thickness must be at least 10 times smaller than the other two dimensions of its corresponding sub domain.

Material data, loads and boundary conditions

On top of the definition of a mesh geometry and its associated physical properties, e.g., sections, thickness, inertias, the FEM requires the definition of material parameters, loads and boundary conditions.

Material data are associated with each finite element in order to generate the global stiffness matrix of the equivalent discretized sub domains representing the initial CAD model. The material properties (homogeneity, isotropy, linearity, ...) are themselves inherent to the model of constitutive law representative of the component's mechanical

behavior. In case of a component made of composite material, the spatial distribution of the different layers of fibers should be carefully represented in the meshed model of this component.

Loads and boundary conditions are essential settings of a mechanical simulation to describe the mechanical effects of other components on the ones of interest. Consequently, loads and boundary conditions are also part of the mechanical simulation pre-processing. A load can be a punctual force applied at a finite element node, a pressure distributed over the surface of a set of finite elements or even a force field, e.g., gravity force. Similarly, boundary conditions have to be attached to a particular set of nodes. Boundary condition settings interact with idealization processes (see Section 1.5.3), e.g., a force acting on an elongated side of a long slender volume is applied to a linear sequence of nodes of the idealized equivalent model defined as a beam model. Consequently, the boundary condition is also dimensionally reduced. In practice, an engineer defines the loads and boundary conditions areas over a component using partitioning operators prior to mesh the component.

1.4.3 FE simulations of assemblies of aeronautical structures

The FEM issues have been extensively addressed for standalone components and integrated in a PDP. However, the FE simulation target is now evolving toward assembly structures, which are under focus in the next section.

An assembly can be regarded as a set of components interacting with each other through their interfaces. These interfaces contribute to mechanical functions of components or sub-assemblies [BKR*12, KWMN04, SLF*13]. An assembly simulation model derives from shape transformations interacting with these functions to produce a mechanical model containing a set of sub domains discretized into FEs and connected together to form a discretized representation of a continuous medium.

Interactions between sub domains in assembly models and associated hypotheses

An assembly simulation model is not just a set of meshed sub domains positioned geometrically in a global coordinate system. These sub domains must be connected to each other to generate global displacement and stress fields over the assembly. To process every assembly interface (see Section 1.3.2), the user should decide which mechanical behavior to apply. Connections between components through their interfaces can be of type kinematic or physical and associated with physical data (stiffness, friction coefficient, ...) and material parameters, as necessary. The selection of connector types is subjected to user's hypotheses regarding the relative behavior of sub domains representing the components, e.g., motion and/or relative interpenetration. Here, a sub domain designates either an entire component or a subset of it when it has been

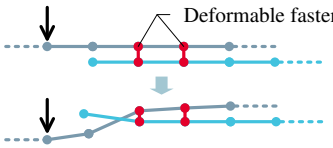
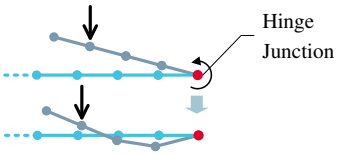
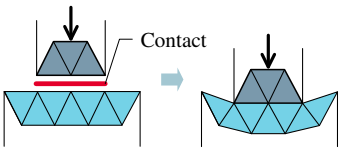
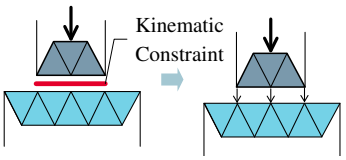
	With relative motion	Without relative motion
With interpenetration	Deformable junctions models: used to model complete mechanical connections with deformable connectors elements, i.e., springs, dampers, ... 	Rigid junctions models: used to model rigid connections, i.e., ball joints, welds, rivets, bolts, ... 
Without interpenetration	Normal and tangential contact: Used to model the normal and tangential stresses (friction) transmitted between two solids in contact during the simulation. 	Kinematic constraints: used to model relationships expressed as displacement/velocity between nodes, e.g., tie constraints, rigid body, ... 

Table 1.2: Connector entities available in CAE software.

idealized. The connection types are synthesized in Figure 1.2.

The introduction in a FEA of a relative motion between components (contacts condition) considerably increases the complexity of this analysis. Indeed, a contact is not a linear phenomenon and requires the use of a specific nonlinear computational model, which slows down the simulation time. Setting up a contact is a strong hypothesis, which leads to the definition of the potential contact areas on both components. The sets of FEs in contact must be carefully specified. On the one hand, they should contain sufficient elements, i.e., degrees of freedom, to cover the local phenomenon while limiting the interpenetration between meshes. On the other hand, they should not contain too many elements to avoid increasing unnecessarily the computation time.

During the early design phases, in addition to idealized models of components, it is common to perform simulations using simplified representation of junctions. In this case, the junction simulation objectives aim at transferring plate loads throughout the whole assembly and FE beam elements are sufficient to model the bolts' behavior. In this configuration, the whole group of components taking part of the junction is replaced by a unique idealized model (see Figure 1.12). When applied to a FEA of large aeronautical structures, these models are called FE connections with fasteners and they are widely used to integrate component interactions with or without contact conditions. A fastener connection may be applied either to mesh nodes or may be

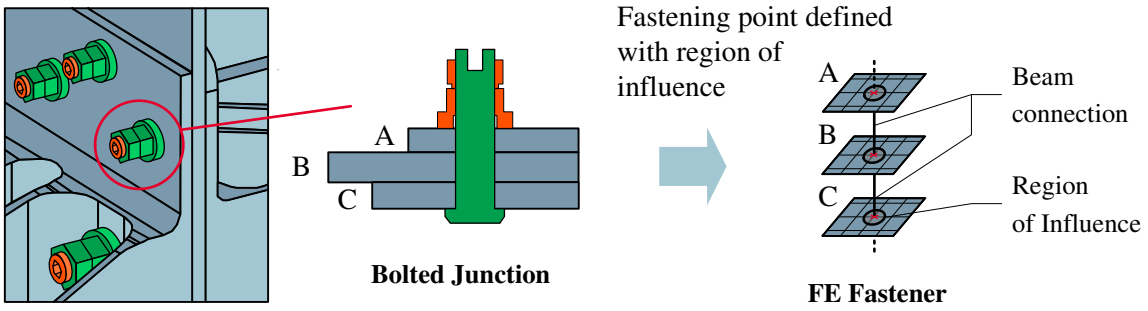


Figure 1.12: Example of a FE fastener simulating the behavior of a bolted junction using beam elements.

mesh-independent, i.e., a point to point connection is defined between surfaces prior to the mesh generation process.

Interactions between simulation objectives and the simulation model preparation process

Simulation objectives drive the shape transformations of CAD solids and interact with the simulation hypotheses to model connections between components. During a PDP, simulations may be used at various steps of a design process to provide different informations about the mechanical behavior of components and sub systems. Based on Troussier's [Tro99] classification, three simulation objectives are taken as examples in Table 1.3 to illustrate how simulation objectives influence the idealization process and the models of interactions between components as part of different simulation models.

As an illustration of the influence of simulation objectives on the generation of different simulation models, Figure 1.13 presents two FE models derived from the same assembly structure of Figure 1.6:

- A simplified model used at a design stage of pre-dimensioning and design choices (see Figure 1.13a). The simulation objective is to estimate globally the load transfer between plates through the bolted junctions and to identify the critical junctions. This model contains idealized components with shell FE in order to reduce the number of degrees of freedom. The junctions are modeled with FE fasteners containing beam elements and a specific stiffness model, i.e., the Huth's law [Hut86]. This model contains 145 000 degrees of freedom and solving it takes 15 minutes, which allows the engineer to test various bolted junctions layouts, part thicknesses and material characteristics;

- A full 3D FEM to validate design choices and check conformity with the certification process prior to physical testing (see Figure 1.13b). The simulation objectives contain the validation of the load transfer distribution among the bolted junctions and the determination of the admissible extreme loads throughout the structure. To adapt the FE model to these simulation objectives while repre-

Element of a simulation process	Pre-dimensioning and design choices	Validation of mechanical tests	Contribution to phenomenon understanding
Simulation objectives	Determine of the number of junctions, a component thickness or material, ...	Analyze the distribution of the stress field in a structure. Locate possible weaknesses.	Understand the behavior of the structure to correlate with results after physical tests
Internal Connections (Interfaces)	Physical junction simplified, no contact (rivet and pin models associated to fasteners).	Physical junction simplified or use of volume patch model. Contact interactions between components.	Complete physical junction, use of volume model with contact interactions.
Components' shape	Large number of components. Idealized: thin parts represented as shell models.	Simplified (shell models) for large assemblies, volume model or mixed dimensional model accepted if rather small number of components.	Small number of components. Complete volume model.
Simulation model	Linear	Linear or nonlinear	Nonlinear

Table 1.3: Examples of interactions or even dependencies between simulation objectives and interfaces as well as component shapes.

senting the physical behavior of the structure, an efficient domain decomposition approach [CBG08, Cha12] uses a coarse 3D mesh (tetrahedral FE) far enough from each bolted junction and a specific sub domain around each bolted junction (structured hexahedral mesh) where friction and pretension phenomena are part of the simulation model. Here, the objective is not to generate a detailed stress distribution everywhere in this assembly but to observe the load distribution areas among bolts using the mechanical models set in the sub domain, i.e., the patch, around each bolt. This model contains 2.110^6 degrees of freedom and is solved in 14 hours. Only one such model is generated that corresponds to the physical test.

Conclusion

This section described the main categories of geometric models used in the FEA of structures. The simulation objectives drive the generation of the simulation models, i.e., FE meshes, boundary conditions, ..., used as input data for solving the FE models. In addition to each component definition, a FE assembly must integrate connection models between meshed components. In Section 1.5, the different modeling hypotheses are analyzed with regard to the geometric transformations applied on the DMU input in order to obtain a new adapted CAD model that can be used to support the generation of a FE mesh model.

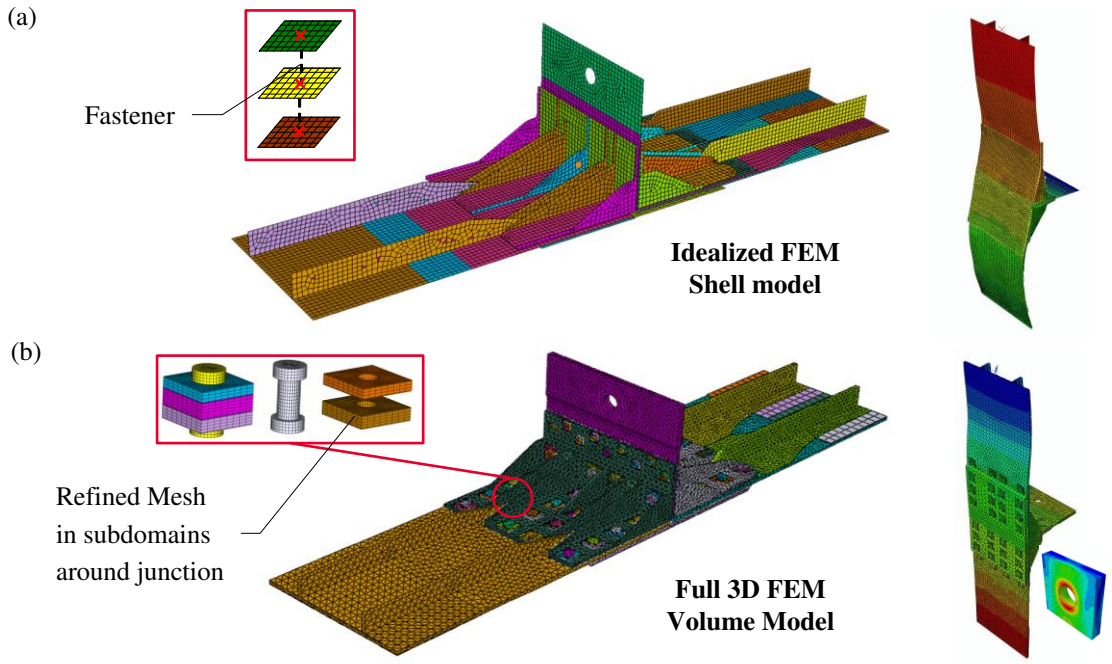


Figure 1.13: Example of aeronautical FE models: (a) an idealized model with fasteners, (b) a full 3D model with a decomposition of plates around each bolted junction and a fine mesh in the resulting sub domain around each bolted junction.

1.5 Difficulties triggering a time consuming DMU adaptation to generate FE assembly models

This section aims at illustrating the complexity of the generation of FE models from DMUs. It highlights the differences between a component's shape in a DMU with respect to the level of abstraction required for a given FEA, especially when a FEA requires an idealization process. This section characterizes and analyzes some specific issues about assembly simulation model preparation and exposes the lack of tools in industrial software, which leads engineers to process manually all the shape transformations and strongly limit the complexity of assemblies that can be processed in a reasonable amount of time.

1.5.1 DMU adaption for FE analyses

Today mechanical structures used in mechanical simulations contain a large number of components, each with a complex shape, binded together with mechanical junctions. In the aeronautic industry, the dimensioning and validation of such structures leads engineers to face two digital challenges:

- The formulation of mechanical simulation models, as developed in Section 1.4,

that can simulate the mechanical behavior of a structure lead to the components' dimensioning as well as the validation of the joint technologies selected (bolting, welding, riveting). During this phase, the engineers have to determine the most adapted simulation model regarding the physical phenomena to observe. They need to set up a FEA and its associated simulation hypotheses that produce the FE model which best meets the simulation objectives with the simulation software environment and technologies available. In practice, a simulation engineer supervises the DMU extraction processes to specify the components to be extracted and/or those having negligible mechanical influences with respect to the simulation objectives. Yet, this assessment is qualitative and is strongly dependent upon the engineer's know-how. Another issue about data extraction stands in the component updates during the PDP. Any geometrical change of a DMU component has to be analyzed by the simulation engineer. Due to the tedious interactive transformations required, a trade-off has to be reached between the time required for the shape update in the FE model and the mechanical influence of the component with respect to the simulation objectives. Here, we face a qualitative judgment;

- The generation of appropriate component shapes from a DMUs to support the generation of simulation models. As explained at Section 1.1, the DMU stands for the geometric reference of a product definition. Through the PLM software, engineers have typically access to DMUs containing the geometry of the 3D assembly defining the product and additional information, essentially about the material properties of each component. However, the extracted DMU representation is not directly suited for numerical FE simulations. Shape transformations are mandatory because designers and mechanical engineers work with different component shapes, resulting in the fact that a DMUs cannot directly support the mesh generation of structural analysis models. These models must meet the requirements of the simulation hypotheses, which have been established when setting up the simulation objectives and the specifications of the mechanical model as part of the FEA. The component shapes generated for the FEA have to be adapted to the level of idealization derived from the specifications of the desired mechanical model, the shape partitioning required for the application of the boundary conditions and loads as well as the level of details of the shape with respect to the FE size required when generating the FE mesh. During the generation of mechanical assembly models, the engineer must also take into account the total number of components, the representation of multiple interfaces between components and a higher level of idealization and larger details than for standalone components, to produce coarse enough assembly models.

To increase the scope of physical assembly simulations, these two challenges lead engineers to use models with simplified 3D representations using idealized shells rather than representations using solids, from a geometric point of view and, from a compu-

tational mechanics point of view, models with specific component interfaces models. These targets require specific treatments during the preparation process of a simulation. Now, the purpose is to describe the major difficulties encountered by engineers during the preparation of assembly simulation models.

1.5.2 Interoperability between CAD and CAE and data consistency

The first difficulty to generate assembly simulation models derives from the interoperability between the CAD and CAE systems. CAD tools have been initially developed in the 60s to help designers modeling solids for applications such as machining or free-form surfaces. CAD has evolved along with CAM (Computer Assisted Manufacturing), driving the functionalities of CAD software. However, simulation software has evolved independently. CAD systems do not support a full native integration of simulation preparation modules. The current practice is to export a DMU to subcontracting companies in charge of the simulation pre-processing, which themselves use specialized CAE software to read and transform the CAD components geometry. Each of these two software, (CAD and CAE), efficiently supports its key process. CAD software are efficient to manage robustly and intuitively modify B-rep solids, to generate large assembly models but they contain basic meshing strategies and most of them are able to model non-manifold objects. CAE software are dedicated to simulation processes, they provide capabilities to describe non-manifold geometry (useful for idealized models) but are limited in modeling non-manifold models. They incorporate robust meshing tools (with topological adaption capabilities) and extensive capabilities to describe contact behaviors, material constitutive laws, ... However, CAE software relies on a different geometric kernel than CAD, which breaks the link between them and leaves open the needs for shape transformation operators.

Also, a transfer of a component from a CAD to a CAE environment has a severe impact on the transferred information. The geometry has to be translated during its import/export between softwares that use different datastructures and operators. This translation can be achieved through a neutral format like STEP (Standard for The Exchange of Product model data) [ISO94, ISO03]. However, this translation may lead to solid model inconsistencies resulting from different tolerance values used in the respective geometric modeling kernels of CAD and CAE software. These inconsistencies may prevent the use of some transformation operators, involving manual corrections. Additionally, the coherence of the input assembly data is crucial. An assembly containing imprecise spatial locations of components and/or components shapes that do not produce consistent CIs (see Section 1.3.2) between components or even the non existence of a geometric model of some components (such as shim components which are not always designed as illustrated in Figure 1.14) implies their manual repositioning or even their redesign to meet the requirements of the simulation model. In the proposed

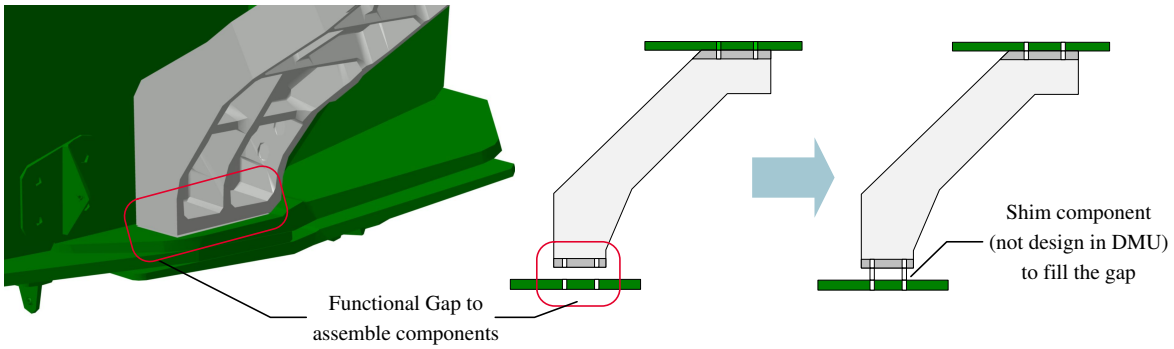


Figure 1.14: Illustration of a shim component which does not appear in the DMU model. Shim component are directly manufacture when structural components are assembled.

approach, the input DMU is assumed to be free of the previous inconsistencies and therefore, it is considered as coherent.

1.5.3 Current operators focus on standalone components

To transform the shape of an initial B-Rep CAD model of a standalone component into a new one as required for its simulation model, the mechanical engineer in charge of the simulation pre-treatment sequentially applies different stages of shape analysis and geometric transformations. His, resp. her, objectives is to produce a new CAD model that can support the mesh generation process. This mesh must be consistent with respect to the simulation objectives and produced in a reasonable amount of time.

Based on the simulation objectives reduced to this component, the engineer evaluates qualitatively and a priori, the interactions between its boundary conditions and its areas of simulation observation, e.g., areas of maximum displacements or maximum stresses, to define whether or not some sub domains of this component should be suppressed, idealized. Currently, engineering practices iteratively apply interactive shape transformations:

1. Idealizations, which are the starting transformations, because they are of highest shape transformation level since they perform manifold dimension reductions;

2. Details removal comes after with topological and skin detail categories [Fin01] that can be also grouped together under the common concept of form feature;

3. Mesh generation requirements leading to solid boundary and volume partitioning are the last step of shape transformations that can be achieved with the so-called ‘virtual topology’ operators or, more generally, meshing constraints [She01, FCF*08].

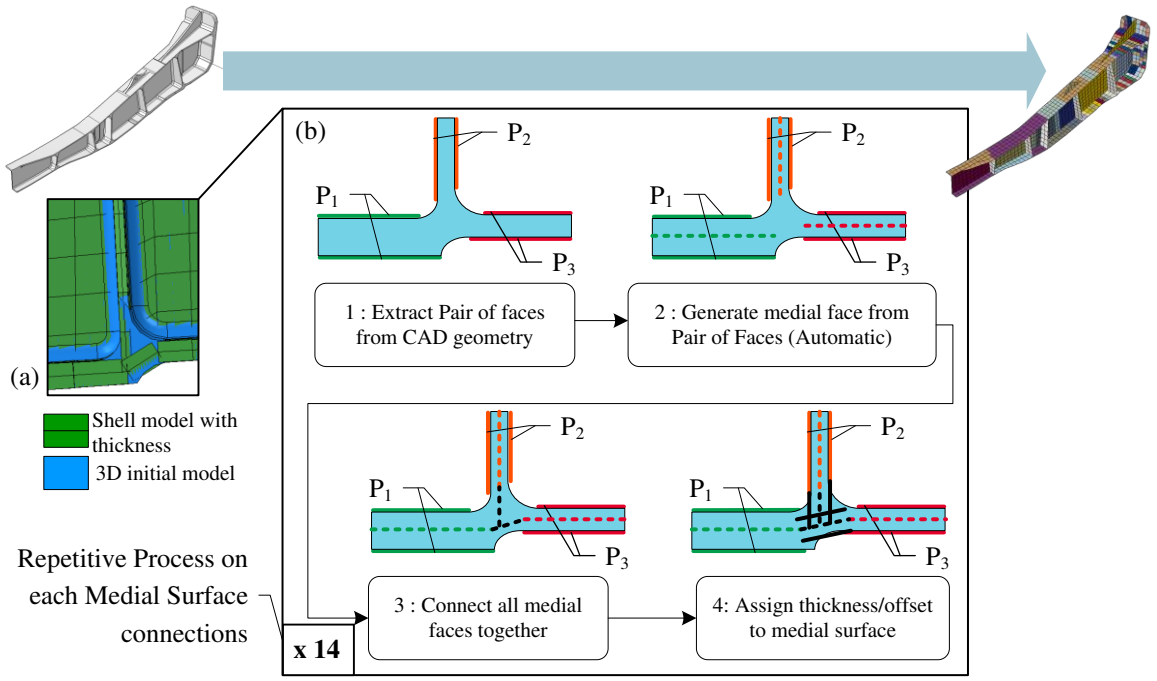


Figure 1.15: Illustration of a manual process to generate an idealized model: (a) initial solid superimposed with its idealized model, (b) iterative process using face pairing identification and mid-surface extensions to connect mid-surfaces.

Commercial softwares already provide some of these operators to adapt DMUs to CAE processes but they are restricted to simple configurations of standalone components. Fewer software, like Gpure [GPu14], offer capabilities to process specific DMU configurations using large facetted geometric models. Component shape transformations, which is the current target of high level operators, are reduced to manual interactions to apply defeaturing operations on CAD parts such as blend removal [ZM02, VSR02], shape simplifications [LAPL05] or to directly remove features on polyhedral models [MCC98, Tau01, LF05]. In all cases, the flow of interactions is monitored by the engineer. This results in very tedious and time consuming tasks requiring a fair amount of resources.

When an idealization is required, engineers can create the resulting mid-surface with a manual and qualitative identification of face pairs [Rez96] or using a medial axis surface generation process [ABD*98, AMP*02]. However, information in between idealizable areas is not available and the engineer has to manually create the connections by extending and trimming mid-surfaces, which is highly tedious and relies also on his, resp. her, mechanical interpretation. Figure 1.15 illustrates a manual idealization process where the user identifies faces pairs, then generates mid-surfaces and creates manually new faces to connect medial faces together while locating the idealized object as much as possible inside the initial volume. Applied to complex shapes, e.g., aircraft structure machined parts, this process flow is highly time consuming as a consequence of the numerous connection areas required and can be hardly automated because slight

shape modifications strongly influence the process flow. Creating idealized domains in areas where face paring cannot be applied rather than leaving a volume domain in these areas, is a common industrial practice to reduce the number of degrees of freedom of a simulation model and reduce the use of mix dimensional models, thus avoiding transfers between shell and volume finite elements because it is not recognized as a good mechanical model. A volume mesh in connection areas is only beneficial if it brings a gain in accuracy, pre-processing or simulation time. Today, generating volume meshes in connection areas requires lots of manual interventions because these volume shapes can be quite complex. Often, the main difficulty is to partition the initial object into simple volumes to generate structured meshes.

The lack of robust and generic operators results in a very time consuming CAD pre-processing task. These geometric operators are analyzed in detail in Chapter 2 to understand why they are not generic and robust enough.

1.5.4 Effects of interactions between components over assembly transformations

The amount of shape transformations to be performed significantly increases when processing an assembly. The engineer has to reiterate numerous similar interactive operations on series of components, the amount of such components being large.

Unlike modeling a standalone component having no adjacent component, an assembly model must be able to transmit displacements/stresses from one component to another. Therefore, the preparation of an assembly model compared to a standalone component implies a preparation process of their geometric interfaces. Consequently, to obtain a continuous medium, the engineer must be able to monitor the stress distribution across components by adding either kinematic constraints between components or prescribing a non-interpenetration hypothesis between them by adding physical contact models. Thus, modeling hypotheses must be expressed by the engineer at each component interface of an assembly.

Today, the interactive preparation of the assembly depicted at Figure 1.13 requires a 5 days preparation to produce either an idealized model or a model based on simplified solids. When looking at this model, some repetitive patterns of groups of components can be observed. Indeed, these patterns are 45 bolted junctions that can be further subdivided into 3 groups of identical bolt junctions, i.e., same diameter. Each group can be further subdivided in accordance with the number of components tightened. The components forming each of these attachments belong to a same function: holding tight in position and transferring forces between the plates belonging to the wing and the fuselage. While a standalone component contributes to a function, an assembly is a set of components that fulfill several functions between them. During an interactive simulation preparation process, even if the engineer has visually identified repetitive

configurations of bolts, he, resp. she, has to transform successively each component of each bolt. A property, by which some components share similar interactions than others and could be grouped together because they contribute to the same function, cannot be exploited because there is no such functional information in the DMU and the geometric models of the components are not structured with their appropriate boundary decomposition to set up the connection with their function, e.g., imprint of contact areas are not generated on each component boundary and the contact areas are connected to a function. Thus, the engineer has to repeat similar shape transformations for each component. However, if the geometric entities contributing to the same function are available, grouped together and connected to their function before applying shape transformations, the preparation process could be improved. For instance, bolted junctions would be located and transformed directly into a fastener model through a single operator. Further than repetitive configurations, it is here the impossibility to identify and locate the components and geometric entities forming these repetitive patterns that reduces the efficiency of the preparation process.

Processing contacts

Hypothesizing the non-interpenetration of assembly components produces non linearity and discontinuities of the simulation model. In this case, the engineer must locate the potential areas of interpenetration during the analysis. Due to the lack of explicit interfaces between components in the DMU, all these contact areas must be processed interactively. At each contact interface, the analyst has to manually subdivide the boundary of each component to generate their geometric interface and then, assign mechanical parameters, such as a friction coefficient, to this interface. In the use-case represented in Figure 1.6, every bolted junction contains between 5 and 7 geometric interfaces at each of the 45 junctions, which amounts to 320 potential contact conditions to define interactively. To avoid these tedious operations, in a context of non linear computations, there is a real need to automate the generation of contacts models in assembly simulations. This automation can be applied to a DMUs with the:

- Determination of geometric interface areas between components, i.e.,
  - Localize geometric interfaces between components likely to interpenetrate during the simulation;
  - Estimate and generate the extent of contact areas over component boundaries. Meshed areas of the two components can be compatible or not depending on the capabilities of CAE software;
- Generation of functional information to set the intrinsic properties of contact models, i.e.
  - Define the friction parameters;

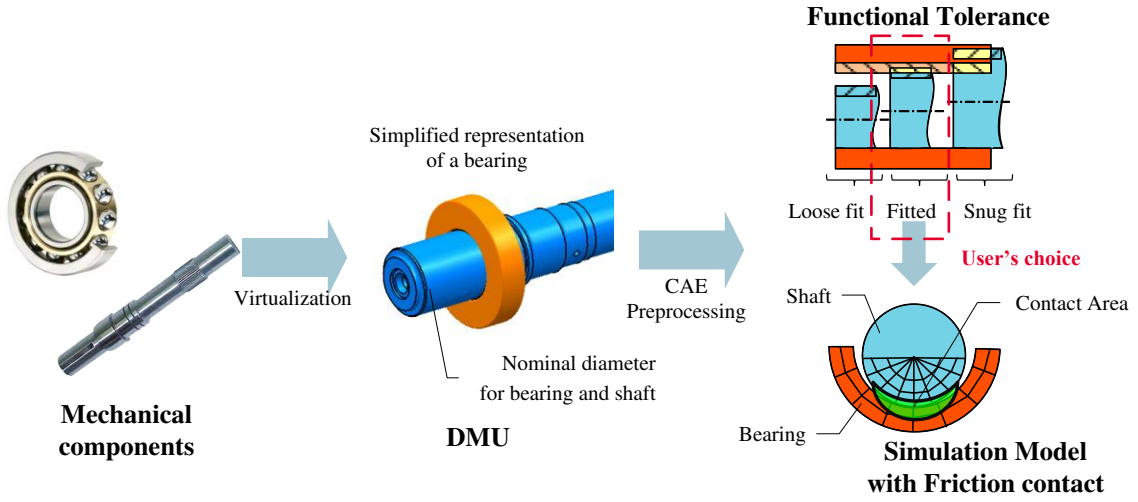


Figure 1.16: Example of contact model for a FE simulation.

– Define the kinematic relations between component meshes in contact areas with respect to the dimensional tolerances between surfaces. Figure 1.16 exemplifies a contact between a shaft and a bearing. Commonly, a DMU exhibits CIs [SLF*12, SLF*13] where components' representations can share the same nominal diameter while they can fulfill different functions according to their fitting (clearance, loose fit, snug fit), thus requiring different settings in their FE respective contact models.

As a result, DMUs do not contain enough information to automate the generation of contact models. FE models need geometric and functional information about components interfaces to delineate contact areas as well as to assign contact model parameters.

Contribution of component functions to the simulation preparation

To automatically handle these repetitive configurations related to components contributing to the same function in an assembly, the simulation preparation process must be able to identify these functions from the input DMU. Currently, the engineer is unable to automate these repetitive tasks because he, resp. she, has no information readily identifying connections in the assembly.

Simulation models chosen by the engineer in a CAE library to replace the junctions are geometrically simple and basic interactive operators are available to achieve the necessary shape transformations. As shown in Figure 1.12, an idealized model of a bolted connection modeled with a fastener consists in a set of points connected by line elements to describe the fastener. Using a mesh-independent fastener, the points representing the centers of the bolt holes in the tightened components do not even need to coincide with a surface mesh node. These idealization transformations are rather simple locally, given the component shapes. Hence, the challenge is neither the

geometric complexity nor the mesh generation. Indeed, it holds in the term 'bolted junction' to identify this geometric set of components and generate geometric relationships between areas of their boundaries. The issue consists in determining the function of each component in an assembly in order to group them in accordance with identical functions and to make decisions about modeling hypotheses (simplification, idealization) on component shapes associated with these identified functions.

Conclusion

Shape transformations taking place during an assembly simulation preparation process interact with simulation objectives, hypotheses and functions attached to components and to their interfaces. To improve the robustness of the geometric operators applied during simulation preparation, and to make them applicable not only to components but also to assemblies, is a first objective to reduce the amount of time spent on assembly pre-processing.

1.6 Conclusion and limits of current practices about DMU manual adaption for FE assembly models generation

Currently, configuring rather complex assembly models for simulations is difficult to handle within the time scale prescribed by an industrial PDP. The pre-processing of CAD models derived from DMUs to produce FE models is far too long compared to the simulation time, it may represent 60% of the whole simulation process (see Section 1.4.1). Consequently, some simulations are not even addressed because their preparation time cannot fit within the schedule of a PDP, i.e., simulation results would be available too late.

Because the shape of CAD models obtained from the engineering design processes is neither adapted to the simulation requirements nor to the simulation solvers, shape transformations are mandatory to generate the simulation models. Consequently, DMUs cannot directly support the preparation process of structural analysis models.

Today, the operators available in CAD/CAE software allow an engineer to perform either interactive geometric transformations leading to very tedious tasks or automated model generation adapted to simple models only or models containing only a restricted set of form features [TBG09, RG03, SGZ10, CSKL04, LF05, LAPL05, ABA02, SSM*10, DAP00]. Unfortunately, these operators are still not generic enough to be adapted to analysts' needs and a rather automated generation of complex component simulation models still raises numerous difficulties, especially when component idealizations must be performed.

To generate assembly simulation models, in addition to its component transformations, the engineers need to generate all the connections between its components. Simulation models for assemblies strongly need geometric interfaces between components to be able to set up boundary conditions between them and/or meshing constraints, e.g., to satisfy conformal mesh requirements. Studying the content and structure of an assembly model, as available in a PDMS, reveals that product assemblies or DMUs are reduced to a set of components located in 3D space without geometric relationships between them. The information about the interfaces between components are generally very poor or nonexistent, i.e., real contact surfaces are not identified or not part of each component boundary. As a consequence, it is common practice for engineers to generate interactively the connections between components, which is error prone, due to the large number of repetitive configurations such as junction transformation.

Finally, processing complex DMUs for the simulation of large assembly models is a real challenge for aircraft companies. The DMUs, used in large industrial groups such as Airbus Group, consist in hundreds of thousands of components. Thus, engineers in charge of such simulations can hardly consider applying the usual methods involving manual processing of all components as well as their interfaces. To meet the needs for large assembly simulation models, improvements in processing DMUs are a real challenge in aircraft companies and it is mandatory to robustly speed up and automate, as much as possible, the DMU transformations required.

1.7 Research objectives: Speed up the DMU preprocessing to reach the simulation of large assemblies

To improve the simulation preparation process of large assembly simulation models, this thesis aims at defining the principles that can be set up to automate the shape adaption of CAD models for the simulation of large assembly structures and developing the associated shape transformation operators. The range of CAD models addressed is not restricted to standalone components but covers also large assembly structures. The tasks planned are mainly oriented toward the transformation of 3D geometric models and the exploitation of their associated semantics for the FEA of structural assemblies applicable to static and dynamic analyses. The task breakdown is as follows:

- Analyze FE simulation rules to extract and classify modeling criteria related to user-defined simulation objectives;
- Based on CAE discipline's rules, specifications and process structure, formalize shape transformations operators to increase the level of automation of component transformations as well as the transformation of its geometric interfaces;

- Implement and validate idealization operators to transform assembly component shapes and assembly interfaces between components while preserving the semantics of the mechanical behavior intended for this assembly;
- Specify the transformation process monitoring and the methodology contributing to the generation of mechanical (CAE) models exploiting a functionally enriched DMUs.

Prior to any automation, a first step outlined in Chapter 2 analyzes in detail the available operators and scientific contributions in the field of data integration and shape transformations for mechanical simulations. The objective is to understand why the current operators and approaches are not robust enough to be applied to aeronautical assemblies. From this analysis, Chapter 3 refines the thesis objectives and exposes a new approach to speed up the shape adaption of CAD assembly models derived from DMUs as needed for FE assembly models. The proposed method is able to adapt a component shape to the simulation objectives and meshing constraints. It incorporates the automation of tedious tasks part of the CAD component idealization process, specifically the treatment of connections between idealizable areas. The proposed algorithms detailed in Chapters 4 and 5 have to be robust, applicable for CAD aeronautical components and preserve the semantic of the mechanical behaviors targeted. These operators contribute to an assembly analysis methodology, presented in Chapter 6, that definitively generalizes assembly transformation requirements in order to prove the capacity of the proposed approach to challenge the generation of large assembly simulation models.

Chapter 2

Current status of procedural shape transformation methods and tools for FEA pre-processing

The transformation of DMUs into structural analysis models requires the implementation of methods and tools to efficiently adapt the geometric model and its associated information. Therefore, this chapter proposes a review of the current CAD-FEA integration related to data integration and shape transformations. In this review, the procedural transformations of CAD components are analyzed, from the identification of details to the dimensional reduction operations leading to idealized representations. The geometrical operators are also analyzed with regard to the problem of assembly simulation preparation. Moreover, this chapter identifies that current geometric operators are lacking application criteria of simplification hypotheses.

2.1 Targeting the data integration level

Chapter 1 described the industrial needs to reduce the time spent on assembly preparation pre-processing for FEA, now the objective of this chapter is to understand why the available procedural geometric modeling methods and operators still do not meet the engineers' requirements, leading them to generate interactively their own models.

Different approaches have been proposed for a better interoperability between CAD and CAE, which can be mainly classified into two categories [DLG*07, HLG*08]:

- Integration taking place at a task level: It refers to the integration of activities of design and structural engineers, hence it relates to design and FEA methodologies and knowledge capitalization in simulation data management;

- Data integration level: It addresses data structures and algorithms performing shape transformations on 3D models of standalone components. More generally, these data structures and operators help connecting CAD and CAE software.

To support the integration of simulation tasks into a PDP, Troussier [Tro99] explains that the knowledge involved in the generation of geometric models is not explicitly formalized. The simulation model definition and generation are based on the collective knowledge of some structure engineers. Therefore, the objective of CAD/CAE integration is not only to reduce the pre-processing time but also to decrease the level of expertise needed to choose and apply the correct transformations to CAD models. Eckard [Eck00] showed that the early integration of structural simulation in a design process could improve a PDP leading to a shorter time-to-market, which applies to assembly processing as well as to standalone components. Badin et al. [Bad11, BCGM11] proposed a specific method of knowledge management used in several interacting activities within a design process. According to them, structure engineers and designers collaborate and exchange design information. However, the authors assume that relationships between dimensional parameters of CAD and simulation models of components are available, which is not necessarily the case. Additionally, they refer to configurations where the shapes of components are identical in both the design and simulation contexts, which is not common practice for standalone components and hardly applicable to assemblies where the reduction of complexity is a strong issue. To help structure engineers, Bellenger [BBT08], Troussier [Tro99] and Peak [PFNO98] formalized simulation objectives and hypotheses attached to design models when setting up simulations. These objectives and hypotheses are subsequently used for capitalization and reuse in future model preparations. This integration at a task level underlines the influence of simulation objectives and hypotheses without setting up formal connections with the shape transformations required.

Since the industrial problems addressed in this thesis focus on the robust automation of shape transformations, it seems appropriate to concentrate the analysis of prior research on the data integration level. These research contributions can categorized in:

- Detail removals performed either before or after meshing a component [LF05, LAPL05, FMLG09, GZL*10];
- Shape simplifications applied to facetted models [FRL00, ABA02];
- Idealization of standalone components [CSKL04, SRX07, SSM*10, RAF11, Woo14] using surface pairing or Medial Axis Transform (MAT) operators.

Section 2.2 analyzes the two first categories of shape simplifications and Sections 2.3 concentrates on the specific transformation of dimensional reduction, which is widely used to generate assembly FEMs as illustrated in Section 1.4. Section 2.4 explores morphological approaches such as the geometric analysis and volume decomposition of 3D solids to enforce the robustness of FE models generation. Finally, Section 2.5

addresses the evolution of the procedural simplification and idealization during component pre-processing from standalone components toward an assembly context.

2.2 Simplification operators for 3D FEA analysis

In CAE applications, the removal of details to simplify a component before meshing it has led to local shape transformations based on B-Rep or polyhedral representations. These transformations create new geometric entities that can incorporate acceptable deviations of a FEA w.r.t. reference results. This section analyzes the different operators and methods aiming at identifying and removing the regions considered as details on 3D solids.

2.2.1 Classification of details and shape simplification

As explained in Section 1.4, the level of detail of a solid shape required for its FE mesh is related to the desired accuracy of its FEA. The removal or simplification of a sub-domain of a solid is valid depending when its associated FEA results meet the accuracy constraint. Armstrong and Donaghy [DAP00] and Fine [Fin01] define details as geometric features which do not significantly influence the results of an FE simulation. Starting from this definition, Fine [Fin01] classifies the details under three categories:

- Skin details: They represent geometric regions which can be removed without changing neither the 3-dimensional manifold property of the solid (see Section 1.2.1) nor its topology (see Section 1.2.2). This category includes the removal of fillets, chamfers, bosses, ...;
- Topological details: This category represents geometric regions which can be removed without changing the 3-dimensional manifold property of the solid but their removal modifies the solid's topology. For example, removing a through hole changes the topology of the solid and the number of hole-loops in the Euler-Poincaré formula decreases;
- Dimensional details: This category represents geometric regions which can be removed and reduce locally the manifold dimension of the solid along with a modification of its topology. This category is related to the idealization process where entire solid models can be represented either with surfaces (dimensional reduction of 1), lines (dimensional reduction of 2) or may even be replaced by a point (dimension reduction of 3).

In this categorization Léon and Fine [LF05] define the concept of detail from a physical point of view. According to them, the result of a FEA can be evaluated with

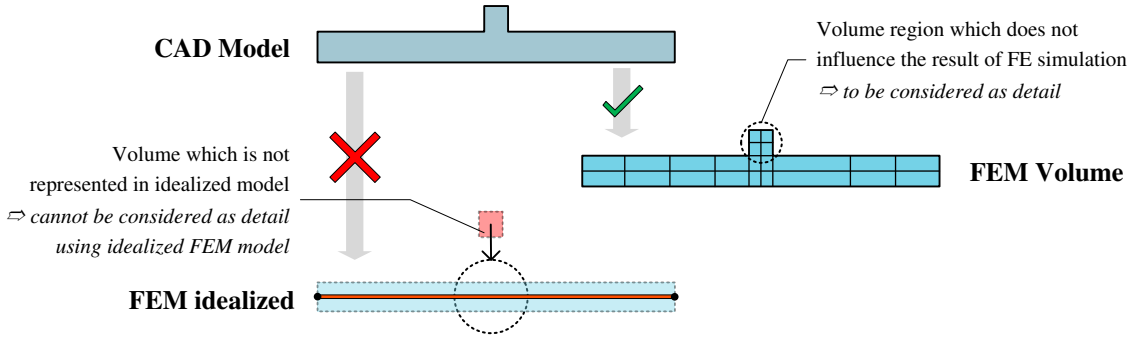


Figure 2.1: Identification of a skin detail related to the accuracy of a FE volume model. With an idealized model, a skin detail cannot be characterized.

'a posteriori error estimators' [EF11, BR78, LL83]. These error estimators characterize the influence of a discretization process, i.e., the FE mesh generation, over the solution of the partial differential equations describing a structure behavior. However, as explained in Section 1.4.3, the behavior simulation of large assemblies is heavily based on idealized models to reduce the size, as much as possible, of simulation models and improve their use during a PDP. In this context, skin and topological details cannot be related to the accuracy of the FEA since the error estimators cannot be applied to shape transformations subjected to a dimensional reduction. Indeed, a volume region which does not satisfy the idealization conditions (see Section 1.4.2) is part of an idealized model but not dimensionally reduced. Therefore, as illustrated in Figure 2.1, evaluating the physical influence of small volume details using an idealized FEM has no meaning because the notion of discretization process is not meaningful over the entire model. When considering idealizations, there is currently no 'error estimators' to evaluate the influence of the dimensional reductions achieved through these transformations. The definition of skin and topological details has to be discussed and extended in the context of dimensionally reduced models.

Even if this classification cannot address idealized models, the simplification operators have to be studied to determine the geometry they are able to process and the information they are able to provide to reduce the complexity of an idealization process. Effectively, it is important to evaluate into which extent skin and topological simplification operators should be applied prior to dimensional reduction or if dimensional reduction takes place first and further simplifications should operate on the dimensionally reduced model. Therefore, the next sections detail the principles of the geometric operators identifying and removing details and determine their suitability to interact with a dimensional reduction process. As mentioned in [Fou07], these operators aim at identifying the geometric regions on the 3D object considered as details and then, remove them from the object in order to generate a simplified model. Approaches to detail removals can be subdivided in two categories depending on the geometric model

describing the component: those which act on tessellated models[1] and those which modify an initial CAD model.

2.2.2 Detail removal and shape simplification based on tessellated models

Although a tessellated object is a simplified representation of an initial CAD model, its geometric model is a collection of planar facets, which can be processed more generically than CAD models. Therefore, the operators based on tessellated models are generic enough to cover a large range of geometric configurations. In what follows, shape simplification operators applicable to the object skin are analyzed first then, a particular attention is paid to the Medial Axis Transform (MAT) operator which extracts an object structure.

Shape simplification

Different approaches have been proposed to simplify the shape of a CAD component using an intermediate faceted model or modifying a FE mesh of this component. These approaches can be synthesized as follows:

- Dey [DSG97] and Shephard [SBO98] improve directly the FE mesh quality by eliminating small model features based on distance criteria compared to the targeted level of mesh refinement. The objective is to avoid poorly-shaped elements and over-densified mesh areas and the treatments proposed are generic;
- Clean-up operators [JB95, BS96, RBO02] repair the degeneracies of CAD models when they have lost information during a transfer between CAD/CAE environments or when they contain incorrect entity connections. Their main issue is the computational cost to recalculate new geometries more suitable for analysis [LPA*03] and the ability of the algorithms to process a wide range of configurations. Furthermore, the geometric transformations are inherently small compared to the model size, which may not be the case for simulation details;
- Others methods [BWS03, HC03, Fin01, QO12] generate and transform an intermediate tessellated model derived from the CAD component. Fine [Fin01] analyses this tessellated geometry using a 'tolerance envelope' to identify and then, remove skin details. Andujar [ABA02] generates new, topologically simplified, models by discretizing the solid object input using an octree decomposition. The advantage of these approaches, dedicated to 3D volume FE, holds in their

[1]Here, it is referred to tessellated models rather than meshes, as commonly used in computer graphics, to distinguish faceted models used in computer graphics from FE meshes that are subjected to specific constraints for FE simulations. Here, the term mesh is devoted to FE mesh.

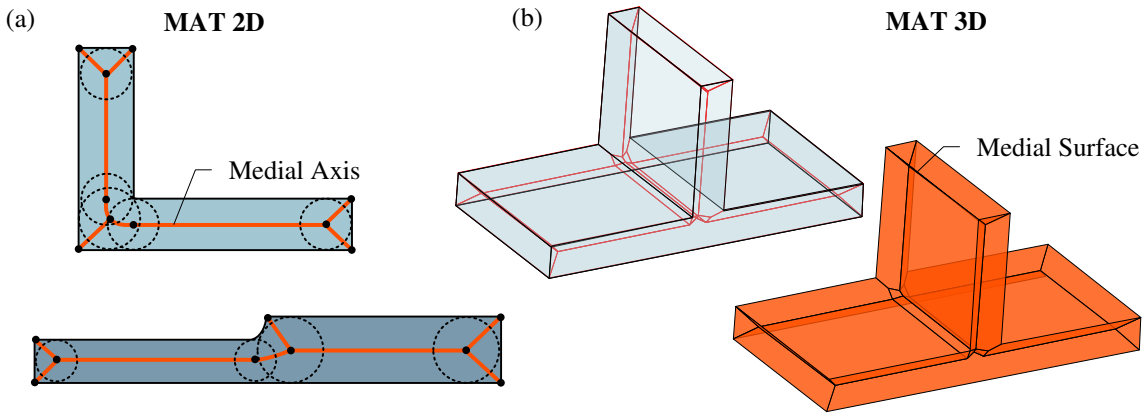


Figure 2.2: Illustration of the MAT: (a) in 2D, (b) in 3D.

independence with respect to the CAD design model. These approaches can support of a wide variety of shapes while avoiding inherent CAD systems issues, i.e., surfaces connections, tolerances, ... Nevertheless, any shape modification of the CAD model cannot be taken into account easily and trigger new iterations of the simplification process;

- Hamri and Léon [OH04] propose an intermediate structure, the High Level Topology, in order to preserve a connection between the tessellated model and the CAD model. As a result, bi-directional mappings can be set between these models, e.g., boundary conditions, B-rep surface types, ... However, the shape transformations are still performed on the tessellated model.

Detail removal using the MAT

To identify shape details in sketches, Armstrong [Arm94, ARM*95] uses the MAT. The MAT has been initiated by Blum [B*67] and represents, in 2D, the shape defined by the locus of centroids of the maximal inscribed circles in a contour (see Figure 2.2a) or, in 3D, by the maximal spheres inscribed in a solid (see Figure 2.2b). The combination of the centerlines and the diameter of the inscribed circle on these centerlines, respectively the center-surfaces in 3D, forms the skeleton-like representation of the contour in 2D, respectively the solid in 3D, called MAT.

As described in [ARM*95], The MAT operator is particularly suited to provide simplification operators with geometric proximity information in 3D and to identify geometric details on planar domains. The authors use a size criterion to identify:

- Details in 2D sketches using the ratio between the length of boundary sketch edges and the radius of the adjacent maximal circle;

- Narrow regions using an aspect ratio between the length of the medial edge to the maximal disk diameter on this local medial edge. A region is regarded as narrow when this ratio is lower than a given threshold. In addition, the authors refer to

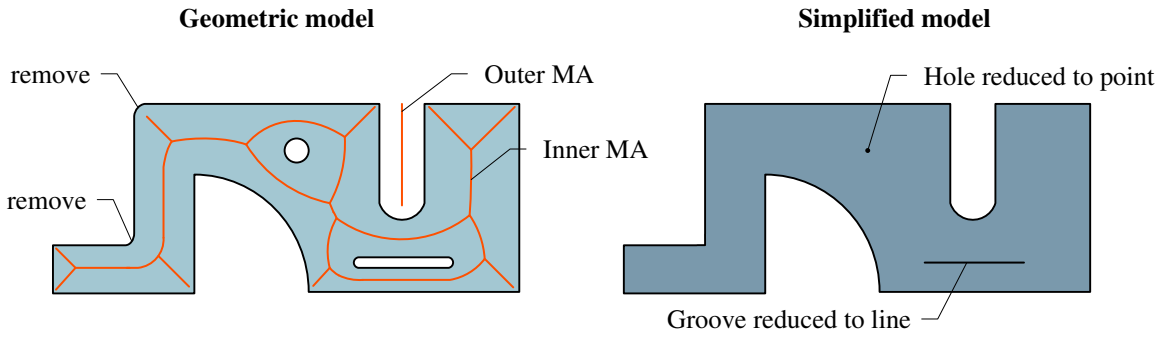


Figure 2.3: Details removal using the MAT and detail size criteria [Arm94].

the principle of Saint-Venant that relates to the boundary conditions location, to categorize a region as a detail.

This method demonstrates the efficiency of the MAT in 2D to analyze, a priori, the morphology of sketch contours. It can compare and identify local regions smaller than their neighborhood. Figure 2.3 illustrates the analysis of a 2D sketch with the MAT [Arm94] to identify details to be removed or idealized. Here, the MAT is addressed as a detail removal operator because the manifold dimension of the 2D domain is not reduced. Nevertheless, it can act also as a dimensional reduction operator. An analysis of the pros and cons of the MAT as a dimensional reduction operator is performed in Section 2.3.1.

Operators based on tessellated models may be applied to a large range of configurations because the input model uses a simple polyhedral definition to represent surfaces in 3D. These operators are efficient to remove skin details before meshing. Yet, large modifications of CAD models are difficult to take into account.

2.2.3 Detail removal and shape simplification on 3D solid models

As explained in the introduction of this chapter, simplifying CAD solids before meshing is a way to enable a robust mesh generation and to obtain directly the shape required for a FEA without local adjustments of the FE mesh. Transformations can be classified into two complementary categories: transformations modifying the boundary decomposition of a B-Rep model without changing the model's shape, transformations modifying the shape as well as its boundary decomposition.

Topology adaption

Virtual topology approaches [SBBC00, She01, IIY*01, LPA*03, ARM*95] have been developed to apply topological transformations to the boundary of an initial B-Rep

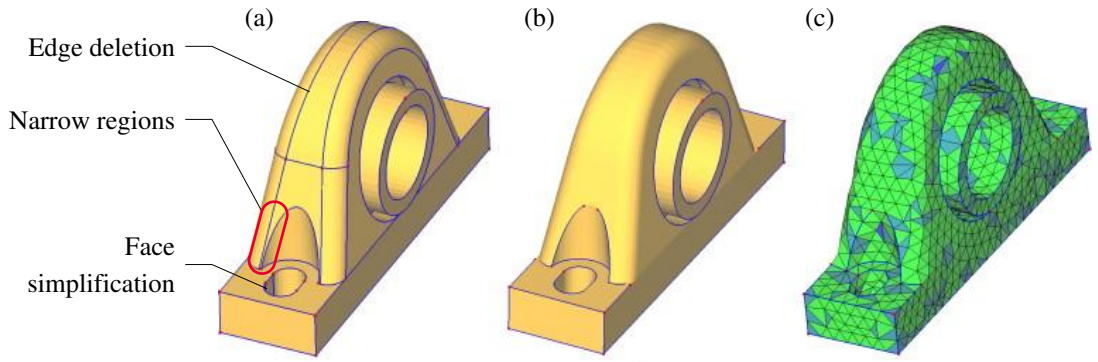


Figure 2.4: Topology adaption of CAD models for meshing [FCF*08]: (a) CAD model, (b) Meshing Constraint Topology obtained with the adaption process, (c) Mesh model generated with respect to Meshing Constraint Topology.

model in order to generate a new boundary decomposition that meet the simulation objectives of this B-Rep model and express the minimum required constraints for mesh generation. Virtual topology approaches belong to the first category of transformations. To anticipate the poorly-shaped mesh elements resulting from B-rep surfaces having a small area, the operation include splitting, merging edges and clustering faces. Anyhow, the objective is to contribute to the generation of a boundary decomposition of a B-Rep model that is intrinsic to the simulation objectives rather being tied to the decomposition constraints of a geometric modeling kernel. Foucault et al. [FCF*08] propose a complementary topology structure called 'Meshing Constraint Topology' with automated adaption operators to enable the transformation of CAD boundary decomposition with mesh-relevant faces, edges and vertices for the mesh generation process (see Figure 2.4). In addition to the topological transformations (edge deletion, vertex deletion, edge collapsing and merging of vertices), the data structure remains intrinsic to the initial object which makes it independent from any CAD kernel representations. Topology adaption is an efficient operator before mesh generation and it is available in most CAE software. However, virtual topology operators are neither generic across CAE software nor they form a complete set of transformations.

Form feature extraction

The main category of solid model simplification is the extraction or recognition of features (holes, bosses, ribs, ...). Different application domains' requirements lead to a wide variety of feature definitions. Here, a feature is defined as in [Sha95] and refers to a primary geometric region to be removed from a B-Rep object and hence, simplifies its shape. The corresponding operators belong to the second category of transformations. The simplification techniques initially define a set of explicit geometric areas identified on an object. Then, specific criteria are applied, for example metrics, to evaluate and select the candidate features to remove. The construction tree resulting from components' design (see Section 1.3) directly provides features that can be evaluated.

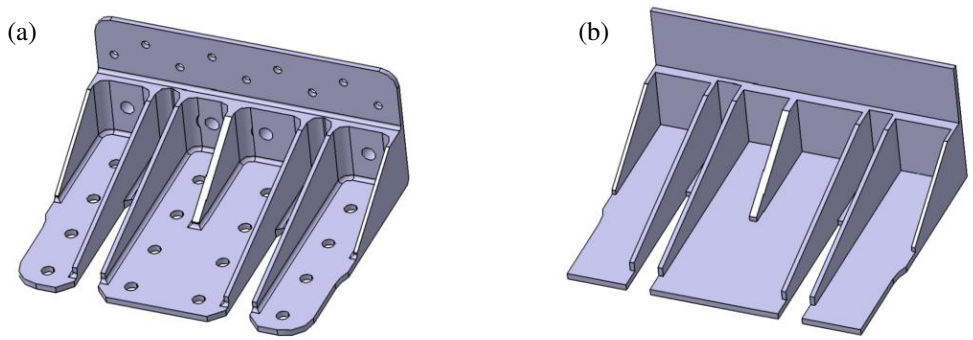


Figure 2.5: Illustration of CAD defeaturing using CATIA: (a) CAD initial model, (b) Simplified CAD model with holes, fillets and chamfers suppressions.

However, this approach relies on the availability of this tree, which is not always the case (see Section 1.5.2 on interoperability). Feature recognition approaches are based on the fact that the construction tree is not transferred from a CAD to a CAE system, and they process directly the B-rep model to recognize features.

A reference survey of CAD model simplification covering feature recognition techniques has been performed by Thakur [TBG09]. For specific discussions on geometric feature recognition see Shah et al. [AKJ01]. A particular domain, mostly studied in the 80-90s is the recognition of machining features. The methods [JC88, VR93, JG00, JD03] in this field are efficient in recognizing, classifying and removing negative features such as holes, slots or pockets. Han et al. [HPR00] give an overview of the state-of-the-art in manufacturing features recognition. Machining feature recognition has been pioneered by Vandenbrande [VR93]. Woo et al. [WS02, Woo03] contributed with a volume decomposition approach using a concept of maximal volume and observed that some of them may not be meaningful as machining primitives. In the field of visualization, Lee et al. [LLKK04] address a progressive solid model generation. Seo [SSK*05] proposes a multi-step operator, called wrap-around, to simplify CAD component. To reduce the complexity of assembly models, Kim [KLH*05] uses this operator and proposes a multi-resolution decomposition of an initial B-rep assembly model for visualization purposes. These operators simplify the parts after detecting and removing small or negative features and idealize thin volume regions using face pairing. Simplification is based on local information, i.e., edge convexity/concavity, inner loops, ... The obtained features are structured in a feature tree depending on the level of simplification. A wide range of shapes is generated with three basic operators. However, the multi-resolution model is subjected to visualization criteria, which may not produce shape transformations reflecting the application of simulation hypotheses, in general. Lockett [LG05] proposes to recognize specific positive injection molding features. Her method relies on an already generated Medial Axis (MA) to find features from idealized models. However, it is difficult to obtain a MA in a wide range of configurations. Tautges [Tau01] uses size measures and virtual topology to robustly identify geometric regions considered as details but is limited to local surface modification.

One common obstacle of feature recognition approaches is their difficulty to set feature definitions that can be general enough to process a large range of configurations. This is often mentioned by authors when features are interacting with each other because the diversity of interactions can lead to a wide range of configurations that cannot be easily identified and structured. In addition, in most cases, the definition of the geometric regions considered as features is based on a particular set of surfaces, edges and vertices extracted from the boundary of the B-Rep object. The assumption is that the detection operations based on the neighboring entities of the features are sufficient to construct both the volume of the features and the simplified object. However, the validity of this assumption is difficult to determine in a general setting, e.g., additional faces may be required to obtain a valid solid, which fairly reduces the robustness of these approaches.

Blend removal

Removal of blends can be viewed as a particular application of features recognition. Automatic blend features removal, and more precisely, finding sequences of blend features in an initial shape, is relevant to FE pre-processing and characterizes shape construction steps. Regarding blends removal, Zhu and Menq [ZM02] and Venkataraman [VSR02] detect and classify fillet/round features in order to create a suppression order for removing these features from a CAD model. CAD software has already proposed blend removal operators and it is these operators that are considered in this thesis (see Figure 2.5 for a example of a CAD component defeaturing result). In general, blend removal can be viewed as a first phase to prepare the model for further extraction and suppression of features.

2.2.4 Conclusion

This section has shown that detail removals essentially address 3D volume simulations of standalone components. The suitability of these simplification operators for assembly structures has not been investigated, up to now. Additionally, the approaches to the automation of detail removal have not been developed for idealization. The definition of details addresses essentially volume domains and refers to the concept of discretization error that can be evaluated with posteriori error estimators. As a result, the relationship between detail removal and idealization has not been addressed. Approaches based on tessellated models produce a robust volume equivalent model but incorporating them with idealization processes, which are often refering to B-Rep NURBS models, does not seem an easy task. Many features-based approaches exist but they are not generic enough to process a wide range of shapes.

The operators presented in this section can be employed in a context of CAD to CAE adaption, provided the areas being transformed are clearly delineated. The difficulty is to determine the relevant operator or sequence of operators in relation to the user

simulation objective. For now, only operators simplifying surfaces of 3D objects have been presented, in the following section idealization operators introduce categories of complexity with the dimensional reduction of standalone components.

2.3 Dimensional reduction operators applied to standalone components

As explained in Section 1.4, to generate idealized models, operators are required to reduce the manifold dimension of 3D solids to surfaces or lines. Different approaches have been proposed to generate automatically idealized models of components for CAE. These approaches can be divided into two categories:

- Global dimensional reduction. These approaches refer to the application of a geometric operator over the whole 3D object, e.g., using the MAT that can be globally applied to this object, to generate an overall set of medial surfaces;
- Local mid-surface abstraction. Mid-surface abstraction addresses the identification of local configurations characterizing individual medial surfaces (using face pairs, deflation) on CAD models and, subsequently, handles ithe connection of these medial surfaces to generate an idealized model.

2.3.1 Global dimensional reduction using the MAT

Armstrong et al. [DMB*96, ABD*98, RAF11] come up with the MAT to generate idealized models from 2D sketches and 3D solids. To identify geometric regions in shell models, which may be represented in an FE analysis with a 1D beam, Armstrong et al. [ARM*95, DMB*96, DAP00] analyze a skeleton-based representation generated with the MAT. Although the MAT produces a dimensionally reduced geometry of an input 2D contour, local perturbations (end regions, connections) need to be identified and transformed to obtain a model suitable for FEA. As for the details identification of Section 2.2, an aspect ratio (ratio of the minimum length between the medial edge and its boundary edges to the inscribed maximum disk along this medial edge) and a taper criterion (maximum rate of diameter change with respect to medial edge length) are computed to automatically identify entities that must be either reduced or suppressed. Based on a user input threshold for aspect ratio and taper, the corresponding areas of the MAT are categorized into regions idealized either as 1D beam element, or regions kept as 2D elements, or regions idealized as 0D element (concentrated mass). Beam ends, that differ from the resulting MAT methodology, are also identified through the topology of the MAT in order to extend the idealizable regions.

More recently, Robinson and Armstrong [RAM*06, RAF11] generalize the approach to 3D solid to identify 3D regions which, potentially, could be idealized as 2D shell

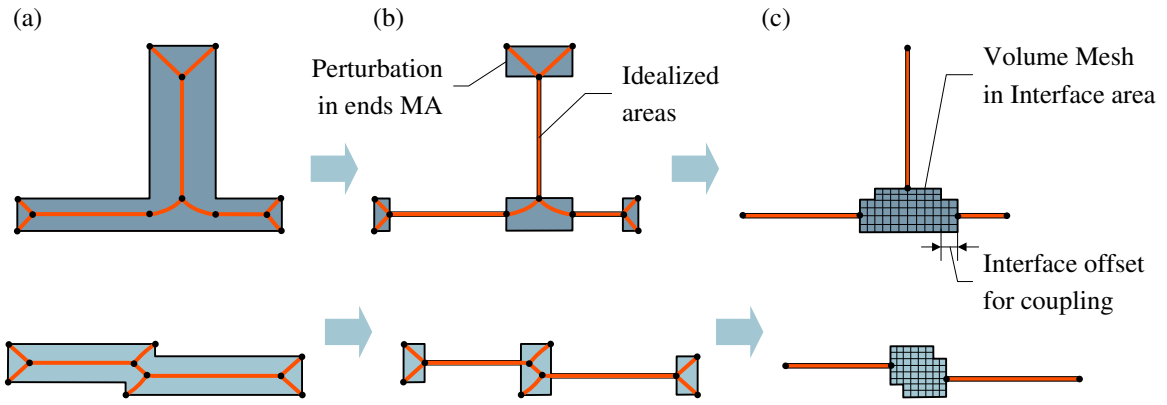


Figure 2.6: Illustration of the mixed dimensional modeling using a MAT [RAF11]: (a) the MAT representation, (b) the model partitioned into thin and perturbations features, (c) the resulting mixed dimensional model.

elements. In a first step, a 3D MAT is used to identify potential volume regions. Then, the MATs of these regions are analyzed by a second 2D MAT to determine the inner sub-regions which fully meet an aspect ratio between their local thickness and MAT dimensions. The final candidates for idealization should satisfy the 2D ratio within the face resulting from the 3D MAT as well as the 3D ratio. Similarly to 2D end regions derived from a 2D MAT, the residual boundary faces from the MAT 3D are extended.

Some limitations of the MAT with respect to idealization processes are:

- The generation of the MAT. Although progress has been made in MAT generation techniques for 3D objects, the computation of an accurate 3D MAT is still a research topic [RAF11]. Even if approaches [CBL11, RG03] exist which enable the computation of a MAT as a G^0 geometric object, 3D MAT from free-form surfaces [RG10, BCL12], B-splines surfaces [MCD11] and planar polyhedra [SRX07], the most efficient algorithms are still based on a discrete representations. The most efficient way to obtain a MAT derives from Voronoi diagrams or from distance fields [FLM03]. An efficient implementation of an algorithm has been proposed by Amenta [ACK01] and, more recently, by Miklos [MGP10]. However, the result is also a discrete representation, which has to be somehow approximated to produce a more global geometric object;

- The need for processing local perturbations (see Figure 2.6b). For mechanical analysis purposes, the topological features in ending regions have to be modified to extend the medial surfaces. These undesirable regions complicate and restrain the analysis domain of the MAT;

- The connection areas. The MAT generates complex configurations in connection areas. Armstrong and Robinson [RAF11] produce mixed dimensional FE models with idealized surfaces or lines and volume domains in the connections between

these regions (see Figure 2.6). These mixed dimensional models, which involve specific simulation techniques using mixed dimensional coupling, do not contain idealized models in connections areas. In addition, to ensure an accurate load transfer from one surface to another, they increase the dimensions of volume connections based on the Saint-Venant's principle (see Figure 2.6c). As a result, the idealized areas are reduced. However, the current industrial practice, as explained in Section 1.4.3, aims at generating fully idealized models incorporating idealized connections. This practice reduces the computational time, reducing the number of degrees of freedom, and ensures a minimum model accuracy based on user's know-how. In this context, the major limit of MAT methods is the processing of these connections areas which do not contain proper information to link medial surfaces.

2.3.2 Local mid-surface abstraction

To generate fully idealized models, alternative approaches to MAT identify sets of boundary entities of the CAD models as potential regions to idealize. Then, mid-surfaces are extracted from these areas and connected together.

Face pairing techniques

Rezayat [Rez96] initiated the research in mid-surface abstraction from solid models. His objective was to combine the geometric and topologic information of the B-rep model to robustly produce idealized models while transforming geometric areas. This method starts with the identification of surfaces which can be paired based on a distance criterion between them. During this identification phase, an adjacency graph is generated representing the neighbouring links between face-pairs. This graph uses the topological relationships of the initial B-rep model. Then, for each face-pair, a mid-surface is generated as the interpolation of this geometric configuration, as illustrated in Figure 2.7a. During the final step,the mid-surfaces are connected together using the adjacency graph of the B-Rep model (see Figure 2.7b). Although this method generates fully idealized models close to manually created ones, the underlying issue is the identification of areas that could potentially be idealized. Indeed, the identification of face-pairs does not ensure that the thickness, i.e., the distance between face-pairs, is as least ten times smaller than the other two directions (see idealization conditions described in Section 1.4.2). The areas corresponding to these face pairs is designated here as tagged areas. In addition, the connection between mid-surfaces requires the definition of new geometric entities which result from an intersection operator. This intersection operator solely relies on the boundary of the areas to be connected, i.e, the face-pairs. There is no information about the boundary of the regions to be idealized as well as the interface areas between their mid-surfaces, e.g., limits of valid connections areas. As illustrated in Figure 2.8d, this information does not appear directly

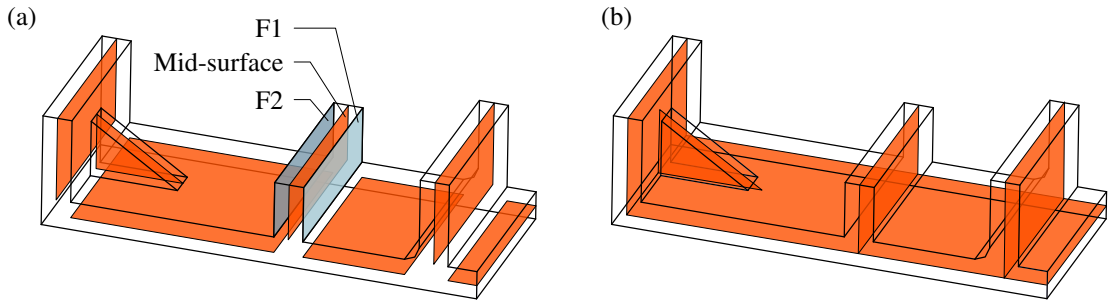


Figure 2.7: Illustration of mid-surface abstraction [Rez96], (a) creation of mid-surfaces from face-pairs, (b) connection of mid-surfaces to generate a fully idealized model.

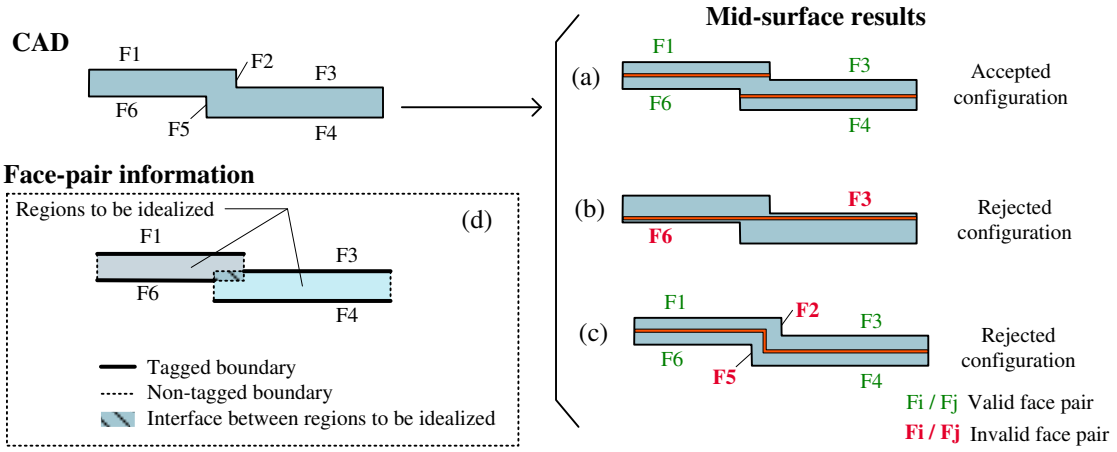


Figure 2.8: An example of particular geometric configuration not addressed by face-pairs methods: (a) Valid configuration without mid-surface connection, (b) and (c) rejection of an invalid face-pair due the overlapping criterion, (d) information on non-tagged areas and interfaces between idealizable regions that are not evaluated with face-pairs methods.

on the initial model. These areas are the complement of tagged areas with respect to the boundary of the initial model; they are named non-tagged areas. As a result, mid-surface abstraction are reduced to specific geometric configurations when the face-pairs overlap each other. As depicted in Figure 2.8, the face-pairs F3-F6 and F2-F5 are rejected due to the overlapping criterion. So, the idealized configurations 2.8b and c are rejected whereas they could be suitable for FEA.

In order to improve the robustness of idealized areas processing, Lee and al. [LNP07a] use a propagation algorithm through the B-rep model topology to identify face-pairs. However, this approach is limited to configurations where the face-pairs can be connected in accordance with predefined schemes. Ramanathan and Gurumoorthy [RG04] identify face-pairs through the analysis of mid-curve relationships of all the faces of the solid model. For each face, its mid-curve is generated using a 2D MAT. This generation is followed by the analysis of the mid-curve graph in order to identify face-pairs. The resulting mid-faces, derived from face-pairs, are then connected to each other in

accordance with the mid-curve adjacency graph. This method increases the robustness of face-pairing, indirectly using the morphology of the paired faces. Analyzing the mid-curve relationships of adjacent faces enables a morphological comparison of adjacent faces. Since mid-curves have been obtained through a MAT, the face-pairs identification depends on the accuracy of this mid-curve generation. This method comes up with face-pairs close to each other and sufficiently large along the two other directions to meet the idealization hypothesis. However, this approach is limited to planar areas.

Negative offsetting operations

Sheen et al. [SSR*07, SSM*10] propose a different approach to generate mid-surfaces: the solid deflation. The authors assume that a solid model can be seen as the result of the inflation of a shell model. Their principle is to deflate the solid model, shrinking it down to a degenerated solid with a minimum distance between faces close to zero. This generates a very thin solid model looking like an idealized model. In a next step, faces are extracted and sewed together to create a non-manifold connected surface model. The issue of this method lies in the generation of the deflated model. Indeed, a face-pairs detection is used to generate the mid-surfaces input to the shrinking operation. This face-pair detection does not cover configurations with a thickness variation, which is common for aeronautical parts and other mechanical components. This approach is similar to a straightforward MAT generation [AA96, AAAG96], which applies a negative offset to boundary lines in 2D, surfaces in 3D, respectively, in order to obtain a skeleton representation. Yet, this representation being an approximation of the MAT, it does not meet everywhere the equal distance property of a mid-surface and does not provide an answer for all polyhedral solids [BEGV08].

2.3.3 Conclusion

As explained in Section 1.4 and 1.5, the shape of a component submitted to a mesh generation depends on the user's simulation objectives. This analysis of dimensional reduction operators highlighted the lack of idealization-specific information to delimit their conditions of application. All geometric regions do not satisfy the idealization conditions and hence, these idealization operators cannot produce correct results in these areas. A purely geometric approach cannot produce directly a fully idealized model adapted to FEA requirements. An analysis process is necessary to evaluate the validity of the idealization hypotheses and determine the boundary and interfaces of the regions to be idealized.

The MAT is a good basis to produce a 3D skeleton structure and provides geometric proximity information between non adjacent faces. However, it is difficult to obtain in 3D and requires post-processing local perturbations and connection areas. Face-pair techniques are efficient in specific configurations, especially for planar objects. Yet, their main issues remain in their validity with respect to the idealization hypotheses and

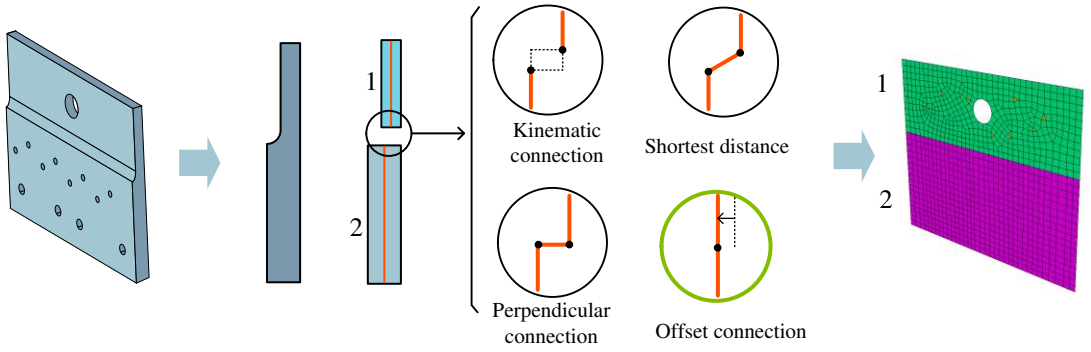


Figure 2.9: Illustration of different connection models for idealized components.

difficulties to process the connection between mid-faces. As illustrated in Figure 2.9, the connection areas derive from specific modeling hypotheses. The user may decide on the connection model that is most appropriate for his, resp. her, simulation objectives.

To improve the dimensional reduction of components, the objectives are expressed as:

1. Identify the volume sub-domains candidate to idealization, i.e., the regions that meet the idealization hypotheses;
2. Obtain additional information on interfaces between sub-domains to generate robust connections there.

2.4 About the morphological analysis of components

As a conclusion of the previous Section 2.3, geometric operators require a pre-analysis of a component shape to determine their validity conditions. Shape decomposition is a frequent approach to analyze and then structure objects. This section aims at studying the operators dedicated to a volume decomposition of 3D objects with an application to FEA.

2.4.1 Surface segmentation operators

There are many methods of 3D mesh[2] segmentation developed in the field of computer graphics. They are mainly dedicated to the extraction of geometric features from these 3D meshes. A comparative study of segmentation approaches of 3D meshes, including

[2]The word mesh is used in the computer graphics context, which refers to a faceted model with no constraint similar to FE meshes.

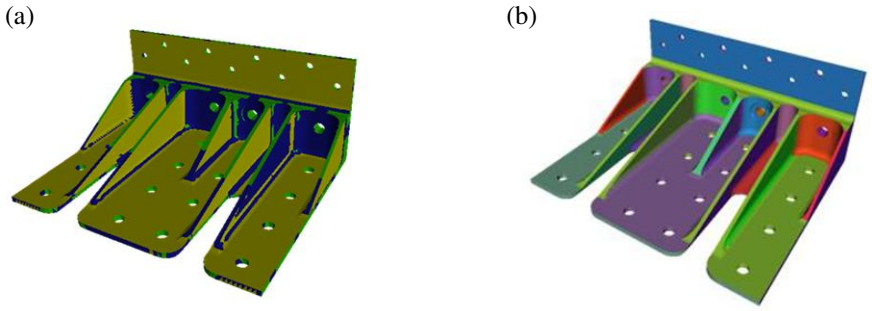


Figure 2.10: Mesh Segmentation: (a) face clustering of Attene [AFS06], (b) shape diameter function of Shapira [SSCO08].

CAD components, is proposed by Attene et al. [AKM*06]. Reference work by Hilaga [HSKK01] applies a Reeb-graph approach to find similarities between 3D shapes. Watershed [KT03, Kos03], spectral analysis [LZ04], face clustering [AFS06], regions growing [ZPK*02, LDB05], shape diameter functions [SSCO08] are other techniques to subdivide a 3D mesh for shape recognition, part instantiation, or data compression. Figure 2.10 illustrates two mesh segmentation techniques [AFS06, SSCO08] on a mechanical part. These algorithms are not subjected to parameterization issues like B-Rep CAD models are. They partition a mesh model into surface regions but do not give a segmentation into volume sub-domains and region boundaries are sensitive to the discretization quality. A post-processing of the surface segmentation has to be applied to obtain volume partitions.

The main objective of the methods cited above is to divide the object in accordance with a "minima rule" principle introduced by Hoffman and Richards [HR84]. This rule consists in dividing this object to conform to the human perception of segmentation. The authors state that human vision defines the edges of an object along areas of high negative curvature. Hence, the segmentation techniques divide a surface into parts along contours of negative curvature discontinuity. In these areas, the quality of an algorithm is based on its ability to meet this "minima rule". Searching for regions of high concavity, algorithms are sensitive to local curvature changes. Depending on the threshold value of extreme curvature, the object may be either over-segmented or under-segmented. Even if this threshold is easier to monitor for CAD components because they contain many sharp edges, the curvature criterion is not related to the definition of idealized areas. Consequently, the methods using this criterion do not produce a segmentation into regions satisfying the idealization hypotheses and regions that can be regarded as volumes.

This section has covered surface segmentation operators that are not appropriate in the context of a segmentation for idealization. The next section studies volume segmentation operators producing directly a decomposition of a solid model into volume partitions.

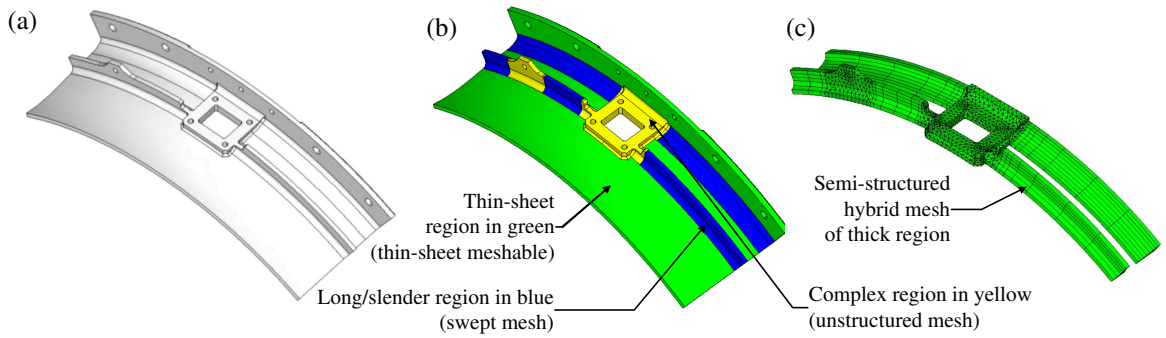


Figure 2.11: Automatic decomposition of a solid to identify thick/thin regions and long and slender ones, from Makem [MAR12]: (a) initial solid model, (b) segmented model, (c) semi-structured hybrid mesh of thick regions.

2.4.2 Solid segmentation operators for FEA

Recently, researches concentrated on the identification of specific regions to automatically subdivide a complex shape before meshing. They address shape transformations of complex parts. The automatic segmentation of a mechanical component into volume regions creates a positive feature decomposition, i.e., the component shape can be generated by the successive merge of the volume regions. This principle particularly applies to dimensional reduction processes, i.e, idealizations.

Volume region identification for meshing

In FEA, solid segmentation methods have been developed to simplify the meshing process. The methods of Lu et al. [LGT01] and Liu and Gadh [LG97] use edge loops to find convex and sweepable sub-volumes for hex-meshing. More recently, the method proposed by Makem [MAR12] automatically identifies long, slender regions (see Figure 2.11). Makem [MAR12] shows that the decomposition criteria have to differ from the machining ones. Heuristics are set up to define the cutting strategy and to shape the sub-domains based on loops characterizing the interaction between sub domains. Setting up these heuristics is difficult due to the large diversity of interactions between sub-domains. Criteria for loop generation aim at generating a unique decomposition and are not able to evaluate alternatives that could improve the meshing scheme.

To reduce the complexity of detecting candidate areas for dimensional reduction, Robinson and al. [RAM*06] use preliminary CAD information to identify 2D sketches employed during the generation of revolving or sweepable volume primitives in construction trees. Figure 2.12 illustrates this process: the sketches are extracted from the construction tree, analyzed with a MAT to determine thin and thick areas forming a feature. Then, this feature is reused as an idealized profile to generate a mixed dimensional model. However, in industry, even if the construction tree information exists in a native CAD model, the creation of features depend on the designer's modeling choices,

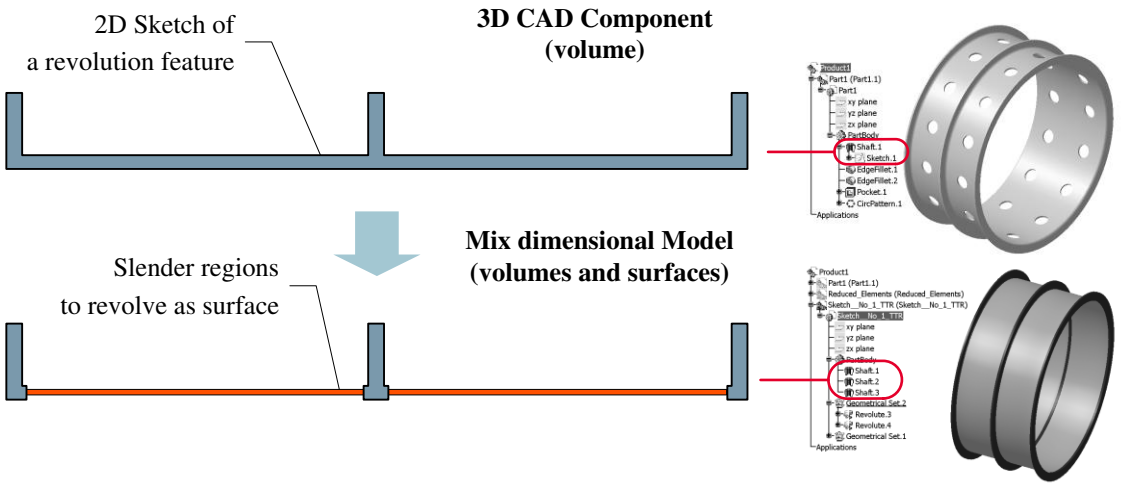


Figure 2.12: Idealization using extruded and revolved features in a construction tree, from [RAM*06].

which do not ensure to obtain appropriate sketches mandatory to get efficient results.

Divide-and-conquer approaches

An alternative to the complexity of the idealization process can be found in divide-and-conquer approaches. Firstly, the solid is broken down into volume sub-domains, which are smaller to process. Then, idealizing these sub-components and combining them together produces the final idealized model.

Chong [CSKL04] proposes operators to decompose solid models based on shape concavity properties prior to mid-surface extractions that reduce the model's manifold dimension. Mid-surfaces are identified from closed loops of split edges and sub-domains are processed using mid-surfaces. The solid model decomposition algorithm detects thin configurations if edge pairs exist in the initial model and matches an absolute thickness tolerance value. Some volume regions remain not idealized because of the nonexistence of edges-pairs on the initial object.

In the feature recognition area, Woo et al. [WS02, Woo03] set a volume decomposition approach using a concept of maximal volume. Their decomposition is based on local criteria, i.e., concave edges, to produce the cell decomposition. Consequently, Woo et al. observe that some maximal volumes may not be meaningful as machining primitives and further processing is required in this case to obtain machinable sub-domains. Recently, Woo [Woo14] describes a divide-and-conquer approach for mid-surface abstraction (see Figure 2.13). A solid model is initially decomposed into simple volumes using the method of maximal volume decomposition [WS02, Woo03] as well as feature recognition of Sakurai [Sak95, SD96]. The mid-surfaces are extracted from these simple volumes using face-pairing. Then, face-pairs are connected using a union Boolean operation, thus creating a non-manifold surface model. Finally, a ge-

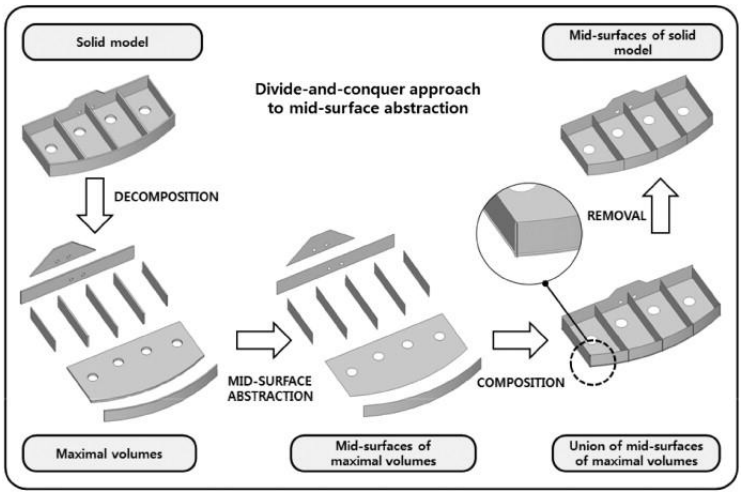


Figure 2.13: Divide-and-conquer approach to idealization processes using a maximal volume decomposition (by Woo [Woo14]).

ometric operator identifies and removes local perturbations of mid-surfaces which do not correspond to the faces of the original model. The major objective of this approach is the complexity reduction of the initial mid-surface abstraction. It increases the robustness of the face paring algorithm by applying it on simpler volumes. However, the connections between mid-surfaces are based on the topology of the initial solid without any analysis of its shape related to the user's simulation objectives. Some solids can be topologically identical but differ in their morphology. Consequently, a morphological analysis of their shape is mandatory to identify the sub-domains subjected to dimensional reduction and to understand the interactions between these sub-domains through their interfaces. Here, the idealization processes are still restricted to a purely geometrical operator that does not integrate user's simulation objectives. Additionally, this method faces difficulties to handle general configurations and connections between idealized sub-domains through mid-surface extension operations.

B-Rep decomposition through feature trees

As observed in Section 2.2, the feature recognition techniques are a way to extract volume sub-domains from B-Rep solids. They support segmentation processes for detail removal but do not provide construction process structures of these B-Rep solids. To this ens, different approaches have been proposed to decompose an object shape into a feature tree.

Shapiro [SV93] and Buchele [BC04] address the B-Rep to CSG conversion as a means to associate a construction tree with a B-Rep model. Buchele [BC04] applies this principle to reverse engineering configurations. CSG tree representations can be categorized into either half-space or bounded solid decompositions. In [SV93, BC04] B-Rep to half-space CSG representation is studied and it has been demonstrated that half-

spaces solely derived from a volume boundary cannot always be integrated into a CSG tree forming a valid solid. In Buchele [BC04], Shapiro and Vossler's approach [SV93] is complemented to generate a CSG representation from scanned objects and to obtain both its geometry and topology. There, additional algorithms must be added to produce complementary half-spaces. Moreover, the meaning of half-space aggregations is not addressed, i.e., there is no connection between the volume faces and primitives that can be used to create it.

Li and al. [LLM06, LLM10] introduce a regularity feature tree used to highlight symmetry in a solid that differs from CSG and construction trees. This tree structure is used to highlight symmetry properties in the object but it neither provides a CSG tree nor primitive entities that could serve as basis for idealization applications. Belaziz et al. [BBB00] propose a morphological analysis of solid models based on form features and B-Rep transformations that are able to simplify the shape of an object and enable simplifications and idealizations. Somehow, this method is close to B-Rep to CSG conversion where the CSG operators are defined as a set of shape modifiers instead of Boolean operators. Indeed, the shape modifiers are elementary B-Rep operators that do not convey peculiar shape information and each stage of the morphological analysis produces a single tree structure that may not be adequate for all simplifications and idealizations.

All the approaches generating a CSG type tree structure from a B-Rep bring a higher level shape analysis with connections to a higher level monitoring of shape transformations, symmetry properties... However, the corresponding framework of B-Rep to CSG conversion must be carefully applied to avoid unresolvable boundary configurations. Furthermore, producing a single tree structure appears too restrictive to cover a wide range of shape transformation requirements.

2.4.3 Conclusion

Solid segmentation operators directly provide a volume decomposition of the initial object. A segmentation brings a higher level of geometric information to the initial B-Rep solid. Previous methods have shown the possibility to generate a segmentation or, even construction processes, from an initial CAD model. Therefore, the current operators:

- do not always produce a complete segmentation, e.g., not all features are identified, and this segmentation is not necessarily suited for idealization due to algorithms focusing on other application areas;
- may be reduced to simple configurations due to a fairly restrictive definition of the geometric areas being removed from the solid. Furthermore, if these operators generate also a construction process, it is restricted to a single process for a component;

- could produce a complete segmentation, e.g., divide and conquer approaches, but they do not ensure that the volume partitions are also simple to process and usable for mid-surfacing.

A more general approach to volume decomposition should be considered to depart from a too restrictive feature definition while producing volume partitions relevant for idealization purposes. Therefore, the difficulty is to find adequate criteria to enable a segmentation for dimensional reduction and connections operators.

The previous sections have presented the main methods and tools for FEA pre-processing and, more specifically, for idealization processes in a context of standalone components. The next section describes the evolution of these operators toward an assembly context.

2.5 Evolution toward assembly pre-processing

Currently, the industrial need is to address the simulation of assembly structures. However, few contributions address the automation of assembly pre-processing. Automated simplifications of assembly for collaborative environment like the multi-resolution approach of Kim [KWMN04] or surface simplification of Andujar [ABA02] transform assembly components independently from each other. This is insufficient to pre-process FE assembly models because mechanical joints and interfaces tightening the different components must take part to these pre-treatment (see Chapter 1.4).

Group of components transformations

In the assembly simplification method of Russ et al. [RDCG12], the authors propose to set component dependencies to remove groups of components having no influence on a simulation, or to replace them by defeatured, equivalent ones. However, the parent/child relationships created from constraint placement of components does not guarantee to obtain the entire neighborhood of a component because these constraints are not necessarily related to the components' geometric interfaces. As explained in Section 1.3.1, these positioning constraints are not necessarily equivalent to the geometric areas of connections between components. Additionally, DMUs don't contain components' location constraints when assemblies are complex, which is the case in the automotive and aeronautic industries to ease design modifications during a PDP. Moreover, part naming identification used in this approach is not sufficient because it locates individual components contained in the assembly, only. Relations with their adjacent components and their associate geometric model are not available, i.e., replacing a bolted junction with an idealized fastener implies the simplification of its nut and its screw as well as the hole needed to connect them in the tightened components.

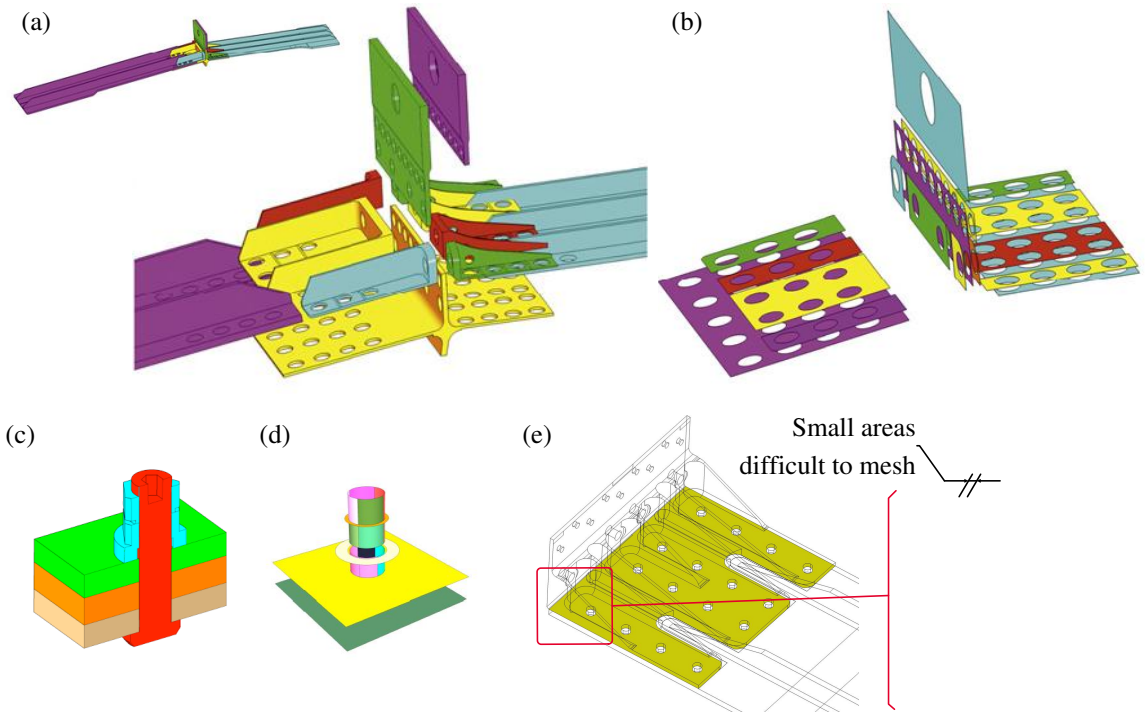


Figure 2.14: Assembly interface detection of Jourdes et al. [JBH*14]: (a) CAD bolted junction with three major plates, (b) some interfaces, (c) cut through of a bolt of this junction, (d) corresponding interfaces, (e) detail of a small geometric area between an interface and the boundary of the component.

Interface detection

To provide mesh compatibility connectivity, i.e., an interface between two components ensures the same mesh distribution in the interface area of each component, Lou [LPMV10] and Chouadria [CV06] propose to identify and re-mesh contact interfaces. Quadros [QVB*10] establishes sizing functions to control assembly meshes. Yet, these methods are used directly on already meshed models and address specific PDP configurations where CAD models are not readily available. Clark [CHE08] detects interfaces in CAD assemblies to create non-manifold models before mesh generation but he does not consider the interactions between interfaces and component shape transformation processes. In [BLHF12], it is underlined that if a component is simplified or idealized, its shape transformation has an influence on the transformation of its neighbors, e.g., a component idealized as a concentrated mass impacts its interfaces with its neighboring components.

Assembly operators available in commercial software are reduced to, when robust enough, the automated detection of geometric interfaces between components. However, their algorithms use a global proximity tolerance to find face-pairs of components and they don't produce explicitly the geometric contact areas. From a STEP representation of an assembly model, Jourdes and al. [JBH*14] describe a GPU approach

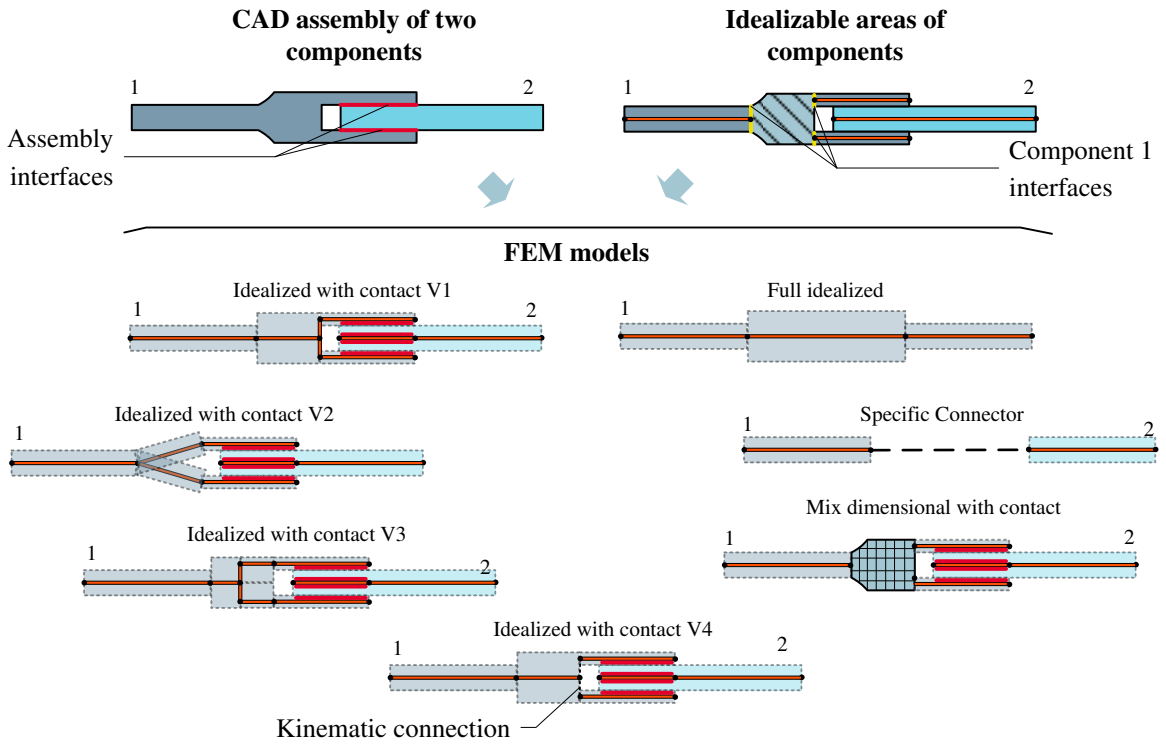


Figure 2.15: Various configurations of the idealization of a small assembly containing two components.

to automatically detect the exact geometric regions of interfaces between components (see Figure 2.14). The results of this technique are used in this thesis as input assembly interfaces data. Yet, obtaining the geometric regions of interfaces is not sufficient, they have to be analyzed to evaluate their suitability with respect to meshing constraints. Figure 2.14e shows the creation of small surfaces, which are difficult to mesh, resulting from the location of an interface close to the boundary of component surfaces.

To reach the requirements of assembly idealizations, the current geometric operators have to take into account the role of assembly interfaces between components, with respect to the shape transformations of groups of components. Figure 2.15 shows the idealization of an assembly containing two components. The idealization of 'component 1' interacts with the idealization of 'component 2', and both interact with the user's choice regarding the assembly interface transformation. Depending on the simulation objectives, the user may obtain either an idealized model of the two components with contact definition, or a globally idealized model from the fusion of the two component. The user may even apply a specific connector model which does not require any geometry other than its two extreme points.

Assemblies bring a new complexity level into idealization processes since the shape of the components and their interactions with their neighbors have to be analyzed before applying geometric operators.

2.6 Conclusion and requirements

The research review in this chapter combined with the context described in Chapter 1 shows that CAD/CAE integration is mainly oriented toward the data integration of standalone components, preparations of assembly models under global simulation objectives require an in-depth analysis and corresponding contributions.

Regarding standalone component processing, although automated operators exist, they are currently effective on simple configurations only. To process complex models, the engineer interactively modifies the component using shape transformation operators according to his/her a priori expertise and evaluation of the simulation model being created. These specific operators, among which some of them are already available in CAE software, have to be selected and monitored by the engineer. Their applications still require lots of manual interactions to identify the regions to transform and correct unintended geometric perturbations. Because of the diversity of simulation contexts, the preconceived idea of applying a generic geometric operator to every component to perform any simplification or idealization, is not valid and must evolve toward simulation context-dependent operators.

The selection of mechanical hypotheses, because of their impact on the DMU model, should also be part of the automation of a mechanical analysis pre-processing. This issue is particularly crucial when performing dimensional reductions on a component. Generating an idealized equivalent model cannot be reduced to the simple application of a dimensional reduction operator. The effects of idealization hypotheses are supposed to have established the connection between the component shape and its simulation objectives. This connection can be made through the identification of geometric areas candidates to idealizations and associated with the connections between idealized sub-domains. An analysis of the component shape, subdividing it into idealizable areas and interfaces between them (see Figure 2.15), is a means to enrich the input CAD solid and prepare the geometry input to dimensional reduction operators. The current volume segmentation operators are restricted to configurations focusing on particular application domains. They often produce a single segmentation into sub-volumes and instantiate the same problem on rather simple configurations. Achieving good quality connections between idealized sub-domains in a robust manner is still a bottleneck of many approaches processing CAD solids for FEA, which requires new developments.

Regarding assembly model processing, there is currently a real lack in scientific research and software capabilities, both. To reach the objective of large assembly structural simulation, pre-processing the DMU, which conveys the product definition, has also to be automated. Assembly simulation models, not only require the availability of the geometric model of each component, but they must also take into account the kinematics and physics of the entire assembly to reach the simulation objectives. This suggests that the entire assembly must be considered when specifying shape transformations rather than reducing the preparation process to a sequence of individu-

ally prepared components that are correctly located in 3D space. As mentioned in Section 1.5.4, to adapt an assembly to FEA requirements, it is mandatory to derive geometric transformations of groups of components from simulation objectives and component functions. As it results from Chapter 1, the knowledge of interface's geometries and additional functional information on components and their interfaces is a good basis to specify these geometry transformation operators on assemblies. In addition, to perform assembly idealizations, structuring geometric models of components in areas to be idealized and component interfaces, is consistent with the assembly structure, i.e., its components and their interfaces. Such a component geometric structure helps preserving the DMU consistency.

Chapter 3

Proposed approach to DMU processing for structural assembly simulations

This chapter sets the objectives of the proposed approach to DMU pre-processing for the simulation of FE structural assembly models. To obtain an efficient transformation of a DMU into a FEM requires geometric operators that process input geometric models which have been structured and enriched with additional functional information. With respect to the objectives set up, the proposed approach uses this enriched DMU, both at 3D solid and assembly levels, to analyze its geometry and connect it to the simulation hypotheses with the required shape transformations.

3.1 Introduction

Chapter 1 has pointed out the industrial context and identified the general problem definition addressed in this thesis. The current practices about model generation for structural mechanical analyses have been described, especially when the resolution is performed using the FE method. Chapter 2 has analyzed the approaches of academia that investigate the automation of FE models generation. The need for shape analysis as a basis of robust geometric transformation operators has been underlined and the lack of research in assembly pre-processing has been pointed out. Figure 3.1 summarizes the manual processes of a DMU transformation for the FEA of assembly structures. The analysis of the ongoing practices has been stated in Section 1.5. A main issue, observed in the aeronautical industry, is the manual and isolated application of geometric transformations on each component of the assembly. An assembly component is considered as a standalone part and the user iterates his, resp. her, global simulation objective on each component as well as on each assembly interface. As a result, the

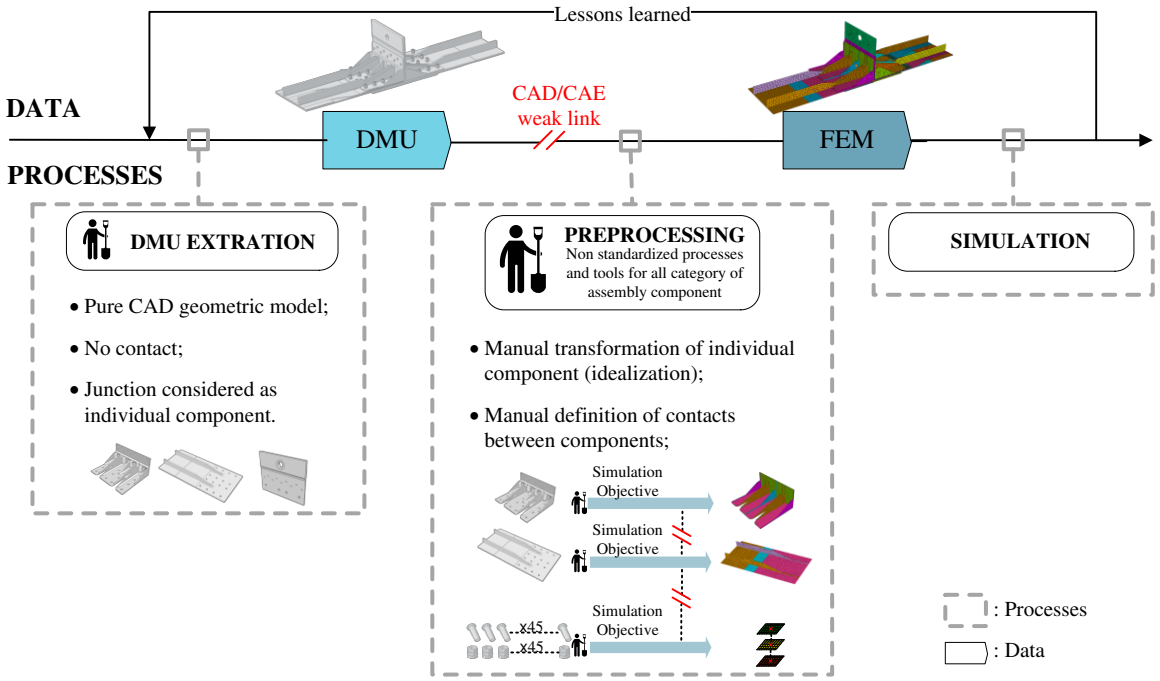


Figure 3.1: Current process to prepare assembly structures. Each component of the assembly is transformed individually.

use of FEA in aeronautical industry is bounded by the time required to set up its associated FEM. Now, the major challenge is the automation of some FEA preparation tasks so that more simulations can be performed on assembly models.

3.2 Main objectives to tackle

As stated in Chapter 2: 'generating an idealized equivalent model cannot be reduced to the simple application of a dimensional reduction operator'. Indeed:

1. Generating simulation models from DMUs requires the selection of the CAD components having an influence on the mechanical behavior the engineer wants to analyze. Setting up the simulation requires, as input, not only the 3D geometric model of component shapes but also their functional information that help selecting the appropriate components (see Section 1.5.4);

2. A DMU assembly is defined by a set of 3D components and by the interactions between them. To automate the preparation of FE assembly models, it is mandatory to take into account the interfaces between components (see Section 1.4.3). An assembly interface, not only contains the geometric information delimiting the contact/interference areas on each component, but contains also the 'functional' information characterizing the behavior of the interface, e.g., clamping, friction, ...;

3. To generate idealized components, i.e., the dimensional reduction process of 3D volumes into equivalent medial surfaces/lines, two main aspects have to be considered:

    - A 3D shape is generally complex and requires different idealizations over local areas depending on the morphology of each of these areas (see Section 2.3);
    - Interfaces between components have an influence on the choice made for these local idealizations. Therefore, the idealization operator has to take into account this information as a constraint, which is not the case of current idealization operators (see Section 2.5).

To address the problem of the FEM preparation of large assembly structures, this chapter introduces an analysis-oriented approach to provide enriched DMUs before geometric transformations. The following sections explain the principles and contributions of this approach, which are subsequently detailed in the next chapters:

- Section 3.3: This section shows that existing approaches are able to provide a certain level of functional information. Two main approaches have been exploited. The method of Jourdes et al. [JBH*14] generates assembly interfaces from DMU models and the method of Shahwan et al. [SLF*12, SLF*13] provides functional designation of components. These methods can be used in our current preprocessing approach to provide enriched DMUs before geometric transformations take place. Nevertheless, some improvements are proposed to take into account the geometric structure of components required for an idealization process;
- Section 3.4: idealization operators necessitate an estimation of the impact of the idealization hypotheses over a component shape, i.e., the identification of areas candidate to idealization. This section sets our objectives to achieve a robust assembly idealization process. They consist in structuring a component's shape and taking advantage of this structure to perform a morphological analysis to identify areas conforming to the user's simulation hypotheses. Subsequently, these hypotheses are used to trigger the appropriate idealization operator over each area;
- Section 3.5: this section outlines the proposed processes exploiting an enriched DMU to robustly automate the major time-consuming tasks of a DMU preparation.

3.3 Exploiting an enriched DMU

Efforts have been made to improve:

- the coordination between engineers in charge of structure behavior simulations and designers;
- the use of simulation results during a design process.

However, as described in Section 1.3, the DMUs automatically extracted from the PLM are not suited for FE simulations. Because of the product structure, DMUs do not contain structural components, only. DMU components have to be filtered during the early phase of FEA pre-processing to avoid unnecessary geometric treatments on components which are considered as details at the assembly level. As explained in Section 1.5.1, this process is based on a qualitative judgment exploiting engineers' know-how. A way to increase robustness of this extraction process, is to have available more useful information for the engineers. At least, this information must contain the functional properties of components.

In addition, considering that the extracted DMU is coherent and contains the exact set of components subjected to shape transformations, the amount of information which can be extracted from the PLM system is not sufficient to set up a robust and automated pre-processing approach to simulations. Even though the extraction of additional meta-data can be improved (see Section 1.5.4), FEM pre-processing requires the exact interface areas between components as well as the functional designation of each component, which are not available in PLM systems, at present.

A main objective of this thesis is to prove that a quantitative reasoning can be made from an enriched and structured DMU to help engineers determining the mechanical influence of components under specific simulation objectives.

Benefiting from existing methods that identify functional interfaces and functional designation of components in DMUs

The method of Jourdes et al. [JBH*14] presented in Section 2.5, detects geometric interfaces between components. Starting from an input B-Rep model, i.e., a STEP [ISO94, ISO03] representation of an assembly, the algorithm identifies two categories of interfaces as defined in Section 1.3.2: surface and linear contacts, and interferences.

The information regarding assembly interfaces are used by Shahwan et al. [SLF*13] to provide, through a procedural way, functional information linked to DMUs.

Even if Product Data Management System (PDMS) technology provides the component with names referring to their designation, this information is usually not sufficient to clearly identify the functions of components in an assembly. For example, in AIRBUS's PLM, a component starting with 'ASNA 2536' refers to a screw of type 'Hi-Lite' with a 16mm diameter. This component designation can be associated under specific conditions to an 'elementary function', e.g., fastening function in the case of 'ASNA 2536'. However, information about each component designation does not iden-

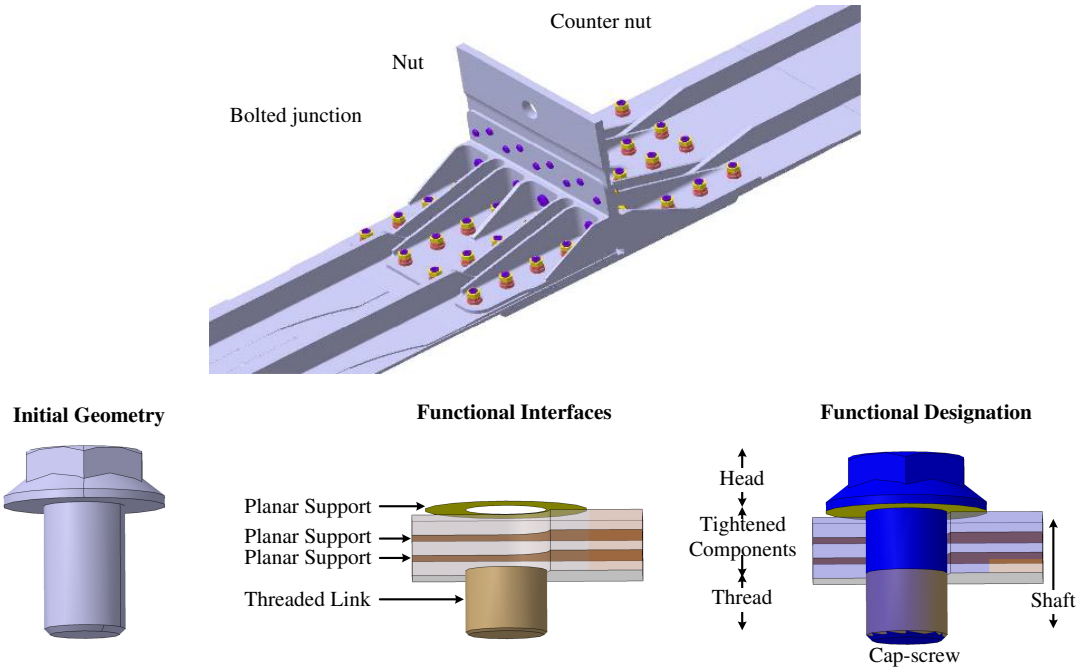


Figure 3.2: Structuring a DMU model with functional properties after analyzing the assembly geometric interfaces and assigning a functional designation to each component (from Shahwan et al. [SLF*12, SLF*13]).

tify its relation with other components inside the scope of a given function, i.e., the geometric model of a component is not structured with respect to its function. How an algorithm can determine which component is attached to another one to form a junction? Which screw is associated with which nut? Additionally, there is a large range of screw shapes in CAD component libraries. How to identify specific areas on these screws through names only? Also, the word screw is not a functional designation; it does not uniquely refer to a function because a screw can be a set screw, a cap screw, ... Therefore, to determine rigorously the functional designation of components, Shahwan et al. [SLF*12, SLF*13] inserted a qualitative reasoning process that can relate geometric interfaces up to the functional designation of components, thus creating a robust and automated connection between 3D geometric entities and functional designations of components. This is a bottom-up approach that fits with our current requirements.

The complete description of a DMU functional enrichment can be found in [SLF*13]. Figure 3.2 shows the result of the functional enrichment of the aeronautical root-joint use-case presented in Figure 1.6. Initially, the DMU was a purely geometric model. Now, it is enriched with the functional designation of components. Using the definition of Shahwan et al. [SLF*13], a functional designation of a component is an *unambiguous denomination that functionally distinguishes one class of components from another.* It relates the geometric model of a component with its functional interfaces (derived from the conventional interfaces described in Section 1.3.2) and with the functional

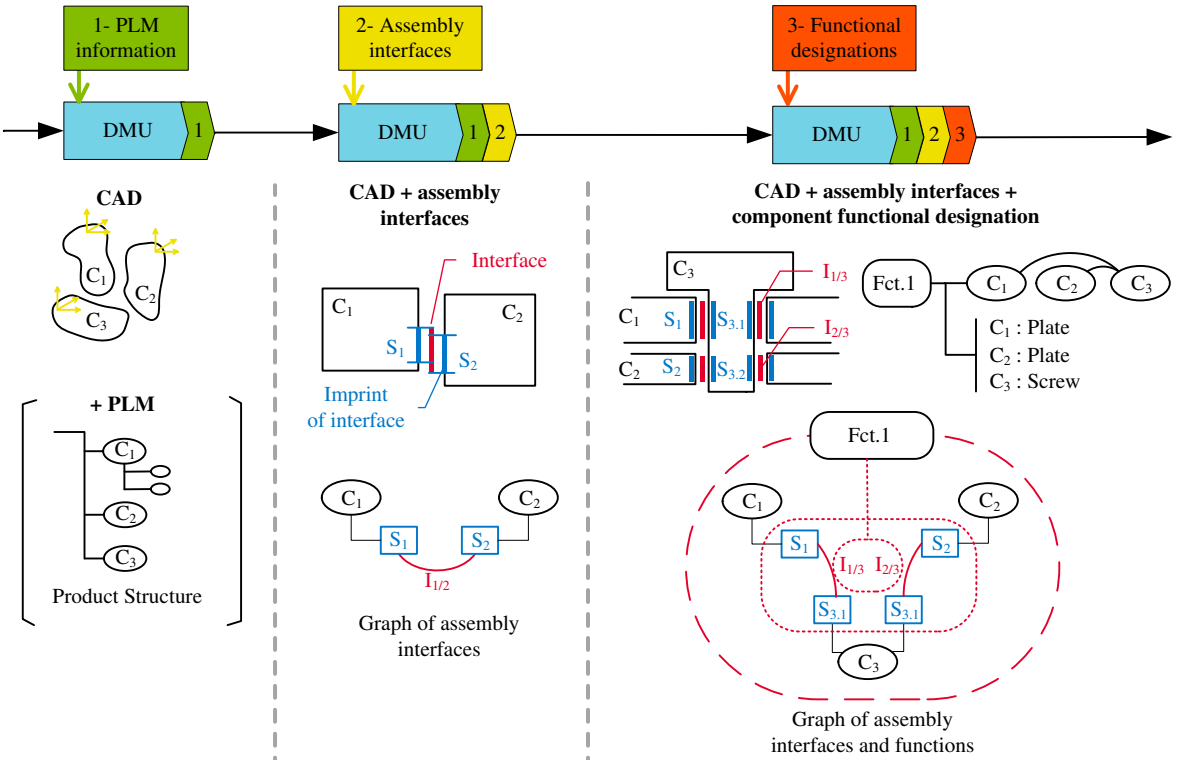


Figure 3.3: DMU enrichment process with assembly interfaces and component functional designations.

interfaces of its functionally related components. The functional designation of a component binds its 3D model and functional interfaces to a symbolic representation of its functions. Regarding screws, illustrative examples of functional designations are: cap screw, locked cap screw, set screw, stop screw, ...

As a result, a component model as well as its geometric model gets structured. As illustrated in Figure 3.3, its B-Rep model contains imprints of its functional interfaces and geometric relationships with functional interfaces of functionally related components. The functional interfaces contain the lowest symbolic information describing the elementary functions of a component and each functional designation expresses uniquely the necessary relations between these elementary functions.

Functional analysis and quantitative reasoning

This enriched DMUs makes available information required to perform a functional analysis of the assembly being prepared for FEA. This analysis allows us to implement a quantitative reasoning which can be used to increase the robustness of the automation of shape transformations of components and their interfaces during an assembly preparation process. Geometric entities locating functional interfaces combined with the functional designation of each component enable the identification and location of groups of components to meet the requirements specified in Section 1.5.4. In the research work described in the following chapters, the fully enriched functional DMU

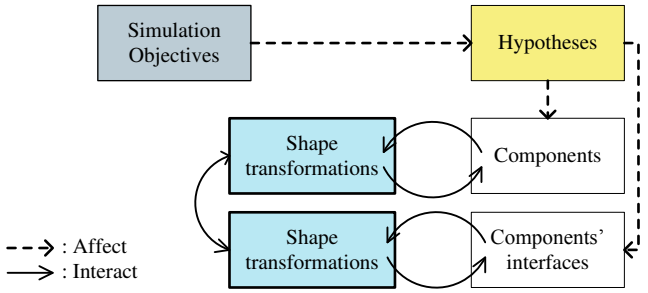


Figure 3.4: Interactions between simulation objectives, hypotheses and shape transformations.

with functional interfaces stands as input data to geometric transformations.

Need to extend the DMU enrichment to a lower level: the component shape

Thanks to a better extraction and functional enrichment of the geometric models of DMU components, new operators are able to identify components and their interfaces that will be subjected to shape transformations. However, this enrichment is not sufficient to determine the geometric areas to be idealized or transformed in the assembly (see Section 2.3). Prior to any shape transformation, we propose to extend this functional assembly enrichment up to the component level. This enrichment is driven by the component's shape and its interaction with the simulation objectives and related hypotheses. The next section highlights the requirements of this enrichment approach.

3.4 Incorporating a morphological analysis during FEA pre-processing

According to Chapter 1, component shapes involved in assembly simulation preparation processes interact with simulation objectives, hypotheses, and shape transformations applied to components and their interfaces. Figure 3.4 shows interactions between shape transformations and FEA modeling hypotheses. To be able to specify the shape analysis tools required, the interactions between shape transformations acting on components as well as on assemblies, on the one hand, and FEA hypotheses, on the other hand, should be formalized. The suggested analysis framework's objective is the reconciliation of simulation hypotheses with geometric transformations.

A morphological analysis driven by idealization needs

As stated in Chapter 2, a morphological analysis dedicated to assembly components can improve the robustness of a geometric idealization process.

The natural way would be to automate the user's approach. Indeed, during the current generation of FE meshes, as explained in Section 1.5, this morphological analysis phase is conducted by the engineer, on each component, individually. Based on his, resp. her, own experience, the engineer visually analyzes the component shape and selects the areas to preserve, to suppress, or to modify. Troussier [Tro99] highlighted the lack of tools helping engineers to build and validate their models. She proposed to refer to previous case studies and make them available to the engineer when a new model has to be built. This knowledge capitalization-based method helps engineers analyze their models through the comparison of the current simulation target with respect to the previously generated simulation models. Indeed, referring to already pre-processed FEM enforces the capitalization principle set up.

However, even if the engineer is able to formalize the simulation hypotheses, one difficulty remains regarding the concrete application of the shape transformations derived from these hypotheses. A visual interpretation of the required geometric transformations, based on past experiences, is feasible for simple components with more or less the same morphology than previous models.

A complex geometry contains numerous regions with their own specific simplification hypotheses. These regions can interact with each other, leading the engineer to reach compromises about the adequate model to generate. For example, many variants of mechanical interactions can appear in interface areas between sub-domains generated for idealized components (see Figure 2.15). It can be difficult for the engineer to get a global understanding of all the possible connections between medial surfaces. As long as no precise mechanical rule exists in these connection areas, each person could have his, resp. her, own interpretation of the hypotheses to apply there. When processing assembly configurations, as illustrated in Section 2.5, its assembly interfaces influence the idealization of the components interacting there. In the case of large assembly structures, on top of the huge amount of time required to analyze all the repetitive configurations, an engineer can hardly anticipate all the interactions between components. Such an interactive analysis process, using the engineer's know-how, does not seem tractable.

Beyond potential lessons learned from previous FEA cases and because current automatic analysis tools are not suited to engineers' needs (see Section 2.4), it is of great interest to develop new automated shape analyzing tools in order to help engineers understand the application of their simplification hypotheses on new shapes. The following objectives derive from this target.

3.4.1 Enriching DMU components with their shape structure as needed for idealization processes

A shape analysis-based approach derives from the information available upstream, i.e., the DMU geometry content before FEA pre-processing that reduces to CAD components. Due to the interoperability issue between CAD and CAE software (see Section 1.5.2), the prevailing practice extracts the B-Rep representation of each component.

During component design, successive primitive shapes, or form features, are sequentially added into a construction tree describing a specific modeling process (see Section 1.2.2). This tree structure is editable and could be analyzed further to identify, in this construction tree, a shape closer to the FE requirements than the final one in order to reduce the amount of geometric transformations to be applied. Often, this modeling process relies on the technology used to manufacture the solid. From this perspective, a machined metal component design process differs, for example, from a sheet metal component one. This difference appears in CAD systems with different workshops, or design modules, targeting each of these processes. As an example, the solid model of a sheet metal component can be obtained from an initial surface using an offset operator. This surface is close to the medial surface that can be used as an idealized representation of this component. However, directly extracting a simpler shape from a construction tree is not a generic and robust procedure for arbitrary components. Such a procedure:

- cannot eliminate all the geometric transformations required to generate a FE model;
- is strongly dependent upon the modeling process of each component;
- and is specific to each CAD software.

Above all and independently of the extracted geometry, it is essential to analyze a component shape before applying it any geometric transformation.

To achieve a robust shape processing, the component shape needs to be structured into regions that can be easily connected to the simulation hypotheses.

Proposal of a volume segmentation of a solid as a component shape structure

Following the recommendations of Section 2.4, we propose to set up a volume segmentation of a 3D solid to structure it as an enriched input model to generate a robust morphological analysis dedicated to mechanical simulations.

As stated in the conclusion of Section 2.4, the generic methods for 3D object decomposition segment mesh[1] models only. Volume decompositions of B-Rep models are

[1] In the computer graphics context.

restricted to specific applications that do not cover the FEA needs.

Here, the objective is set on a proposal of a robust segmentation of a B-Rep solids to enrich them. Because the robust generation of quality connections between idealized sub-domains is still a bottleneck of many approaches that process CAD solids for FEA, the proposed segmentation should incorporate the determination of interfaces between the volumes resulting from the segmentation. The proposed method is based on the generation of a construction graph from a B-Rep shape. This contribution is detailed in Chapter 4.

3.4.2 An automated DMU analysis dedicated to a mechanically consistent idealization process

An analysis can cover multiple purposes: physical prediction, experimental correlation,... The proposed analysis framework is oriented toward the geometric issues about the idealization of assemblies for FEA. Section 2.3 has revealed that a major difficulty encountered by automated methods originates from their lack of identification of the geometric extent where simplification and idealization operators should be applied.

Using the new structure of a component shape, our objective is placed on a morphological analysis process able to characterize the idealization transformations that can take place on a component shape. Therefore, this process should incorporate, during the pre-processing of DMU models, a set of operators that analyze the initial CAD geometry in order to connect it to simulation hypotheses and determine the geometric extent of these hypotheses. Chapter 5 is dedicated to this contribution. The objectives of this morphological analysis enumerate:

- The identification of regions considered as details with respect to the simulation objectives. The DMU adapted to FEA should contain only the relevant geometric regions which have an influence on the mechanical behavior of the structure;

- The identification of relevant regions for idealization compared to regions regarded as volumes. The morphology of a component has to be analyzed in order to determine the thin regions to be transformed into mid-surfaces and the long and slender regions to be transformed into beams. Also, this morphological analysis has to provide the engineer with a segmentation of components into volume sub-domains which have to be expanded into the whole assembly;

- The characterization of interfaces between idealizable regions. These interfaces contain significant information regarding the interaction between idealizable regions. They are used to connect medial surfaces among each other.

Finally, the DMU enrichment process is completed with assembly information as well as information about the shape structure of each component. Consequently, this enriched DMU is geometrically structured. It is now the purpose the next section to carry on setting up objectives to achieve a robust pre-processing from DMU to FEM.

3.5 Process proposal to automate and robustly generate FEMs from an enriched DMU

Figure 3.5 summarizes the corresponding bottom-up approach proposed in this thesis. The first phase uses the methods Jourdes et al. [JBH*14] and Shahwan et al. [SLF*12, SLF*13] to enrich the DMU with assembly interfaces and functional designations of components as recommended in Section 1.5.4. The initial CAD solids representing components are also enhanced with a volume decomposition as suggested in Section 2.4 to prepare a morphological analysis required to process the idealization hypotheses.

The second phase analyses this newly enriched DMU to segment it in accordance with the engineer's simulation objectives (see Section 3.4), i.e., to identify areas that can be idealized or removed when they are regarded as details. This results in the generation of a so-called contextualized DMU.

Providing the engineer with a new contextualized DMU does not completely fulfill his, rep. her, current needs to create geometric models for structural analysis. Consequently, the proposed scheme should not only develop and validate methods and tools to structure and analyze a DMU up to its component level, but also contain processes to effectively generate FE assembly models. In the third phase, the functional and morphological analyses lead to the definition of the assembly transformation process as planed in the second phase, i.e., the transformation of groups of components including dimensional reduction operations.

Exploiting the contextualized DMU, it is proposed to develop a two level adaption process of a DMU for FEA as follows:

- One process is dedicated to standalone geometric component idealization. The objective of this new operator is the exploitation of the morphological analysis and hence, to provide the engineer with a robust and innovative approach to 3D shape idealization;

- Another process extending the idealization operator to assembly idealization. This operator is a generalization of the standalone operator adapted to assembly transformation requirements. To implement this process, we set up a generic methodology taking into account the simulation requirements, the functional assembly analysis and assembly interfaces.

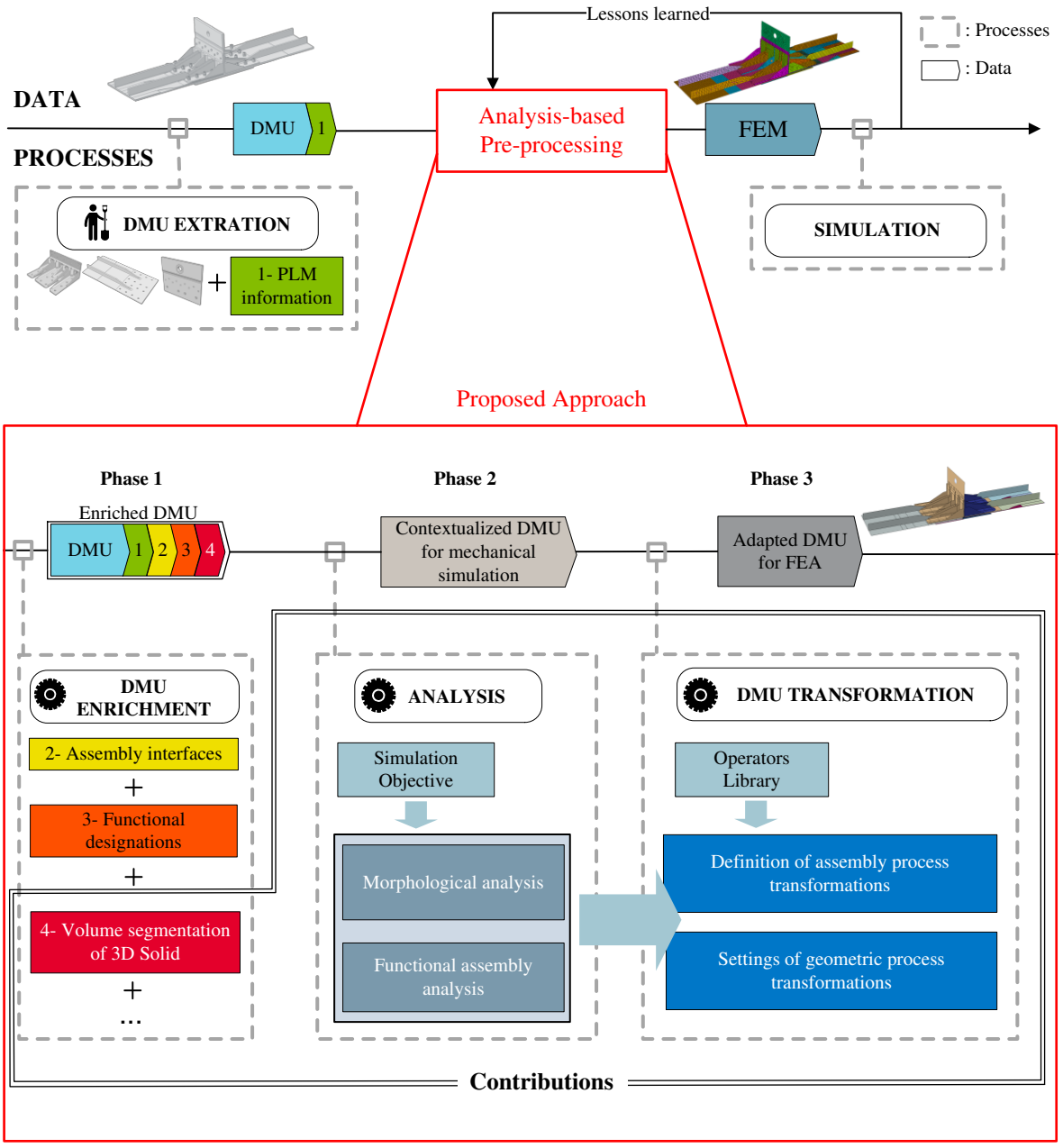


Figure 3.5: Proposed approach to generate a FEM of an assembly structure from a DMU.

3.5.1 A new approach to the idealization of a standalone component

When components have to be fully idealized, their pre-processing requires the development of a robust idealization process containing a dimensional reduction operator associated with a robust one that connects medial surfaces. As shown in Section 2.3, existing approaches face two issues:

- The extraction of a mid-surface/medial line from an idealized sub-domain. Current dimensional reduction operators focus directly on the generation of mid-surface/medial line without having completely evaluated the idealization hypotheses and determined the sub-regions associated to these hypotheses;
- The connection of the set of extracted mid-surfaces/medial lines. Current operators encounter difficulties to generate consistent idealized models in connections areas, i.e., regions which usually do not satisfy the idealization conditions.

To cover these issues, we propose to analyze the morphology of the shape before applying a dimensional reduction operator. Therefore, this operator focuses on the extraction of medial surfaces only in the sub-domains morphologically identified as plate/shell models and on the extraction of medial lines in the sub-domains morphologically identified as beam models. Simultaneously, this morphological analysis is used to provide information on internal interfaces between sub-domains to be idealized. We propose to exploit this new information within the idealization operator to produce consistent geometric models, i.e., on-purpose idealization of sub-domains with on-purpose connections between them. This process is detailed in Section 5.5.

3.5.2 Extension to assembly pre-processing using the morphological analysis and component interfaces

The second process required addresses the transformation of assembly models. The proposed operators have to be applicable to volume sub-domains, which can originate from components or from a group of components.

Evolving the concept of details in the context of assembly structures

Section 2.2 has shown that the relationship between detail removal and idealization processes has not been investigated. The definition of details stated in [LF05, RAF11] addresses essentially volume domains and refers to the concept of discretization error that can be evaluated with a posteriori error estimators.

Assemblies add another complexity to the evaluation of details. It is related to the existence of interfaces between components. As illustrated in Section 1.5.4, interfaces

are subjected to hypotheses to define their simulation model and Table 1.2 points out the diversity of mechanical models that can be expressed with simulation entities. Recently, Bellec [BLNF07] described some aspects of this problem. Yet, comparing the respective influences of rigid versus contact interface models is similar to the evaluation of idealization transformations: this is also a complex issue.

The concept of detail, apart from referring to the real physical behavior of a product, is difficult to characterize for assembly idealization. The structural engineer's know-how is crucial to identify and remove them with interactive shape transformations. Benefiting from the morphological analysis of components' shapes, another objective of this thesis is to provide the user with tools that show areas that cannot be regarded as details (see Sections 5.3.1 and 5.5.2). This way, the concept of details evolves from standalone component to assembly level pre-processing.

Automated transformations of groups of components

As explained in Section 1.5.4, the transformation of groups of components, e.g., junctions, by pre-defined FE simplified geometry, e.g., fasteners, is a top requirement to reduce the FEM preparation time.

Focusing on these specific configurations, the main issue remains the robust identification of the components and assembly interfaces to be transformed. Another objective of this thesis is also to provide a robust operator to identify and transform configurations of groups of components involved in the same assembly function, which is detailed in Chapter 6.

From the analysis of DMU transformation requirements for FE assembly model preparation [BLHF12], the proposed method relies on a qualitative reasoning process based on the enriched DMU as input. From this enriched model, it is shown that further enrichment is needed to reach a level of product functions where simulation objectives can be used to specify new geometric operators that can be robustly applied to automate component and assembly interface shape transformations. To prove the validity of this approach, Section 6.3 presents a template-based operator to automate shape transformations of bolted junctions. The method anticipates the mesh generation constraints around the bolts, which also minimizes the engineer's involvement.

3.6 Conclusion

This chapter has introduced the main principles and objectives of the proposed analysis-oriented approach of DMU pre-processing for the simulation of FE structural assembly models. This approach covers:

- The enrichment of the input geometry, both at 3D solid and assembly levels.

It is critical to provide a volume decomposition of the geometric model of each component in order to access and fully exploit their shape. This structure is a good starting point for the identification of areas of interest for idealization hypotheses. At the assembly level, the DMU is enriched with geometric interfaces between its components, i.e., contacts and interferences, and with the functional designation of components;

- The development of an analysis framework for the simulation of mechanical structures. From the enriched DMU model, the analysis framework can be used to specify geometric operators that can be robustly applied to automate component and interface shape transformations during an assembly preparation process. In accordance with the context of structural simulations, this framework evaluates the conditions of application of idealization hypotheses. It provides the engineer with the operators dedicated to shape adaption after idealizable volume sub-domains have been identified. Also, after the areas considered as details have been identified and information about sub-domains interfaces have been added, the user's modeling rules can be applied in connection areas;

- The specification of geometric operators for the idealization of B-rep shapes and operators transforming groups of components, such as bolted junctions, benefiting from the previously structured DMU. Through the development of such operators, the proposed approach can be sequenced and demonstrated on aeronautical use-cases.

The next chapters are organized in accordance with the proposed approach, as described in the current one. Chapter 4 details the B-Rep volume decomposition using the extraction of generative construction processes. Chapter 5 describes the concepts of the FEA framework using a construction graph to analyze the morphology of components and derive idealized equivalent models. Chapter 6 extends this approach to a methodology for assembly idealization and introduces a template-based operator to transform groups of components.

Chapter 4

Extraction of generative processes from B-Rep shapes to structure components up to assemblies

Following the global description of the proposed approach to robustly process DMUs for structural assembly simulation, this chapter exposes the principles of the geometric enrichment of components using a construction graph. This enrichment method extracts generative processes from a given B-Rep shape as a high-level shape description and represents it as a graph while containing all non trivial construction trees. Advantageously, the proposed approach is primitive-based and provides a powerful geometric structure including simple primitives and geometric interfaces between them. This high-level object description is fitted to idealizations of primitives and to robust connections between them and remains compatible with an assembly structure containing components and geometric interfaces.

4.1 Introduction

Based on the analysis of DMU transformation requirements for FE assembly model preparation in Chapter 1 as well as the analysis of prior research work in Chapter 2, two procedures are essential to generate the mechanical model for the FEA of thin structures:

- The identification of regions supporting geometric transformations such as simplifications or idealizations. In Section 1.4.2, the analysis of thin mechanical shell structures introduces a modeling hypothesis stating that there is no normal stress in the thickness direction. This hypothesis is derived from the shape of

the object where its thin volume is represented by an equivalent medial surface. The idealization process connects this hypothesis with the object shape. Section 2.3 illustrates that idealization operators require a shape analysis to check the idealization hypothesis on the shape structure and to delimit the regions to be idealized;

- The determination of interface areas between regions to be idealized. Section 2.3 showed that the interface areas contain the key information to robustly connect idealized regions. In addition to idealizable areas, the determination of interfaces is also essential to produce fully idealized models of components.

The proposed pre-processing approach, described in Chapter 3, is based on the enrichment of the input DMU data. More precisely, the B-rep representation of each CAD component has to be geometrically structured to decompose the complexity of their initial shape into simpler ones. At the component level, we propose to create a 3D solid decomposition into elementary volume sub-domains. The objective of this decomposition is to provide an efficient enrichment of the component shape input to apply the idealization hypotheses.

This chapter is dedicated to a shape decomposition method using the extraction of a construction graph from B-Rep models [BLHF14b, BLHF14a]. Section 4.2 justifies the extraction of generative construction processes suited for idealization processes[1]. Section 4.3 sets the modeling context and the hypotheses of the proposed approach. Section 4.4 describes how to obtain generative processes of CAD components, starting from the identification of volume primitives from a B-Rep object to the removal process of these primitives. Finally, Section 4.5 defines the criteria to select the generative processes generating a construction graph for idealization purposes. This construction graph will be used in Chapter 5 to derive idealized models. In a next step, the component perspective is extended to address large assembly models. Consequently, the segmentation approach is analyzed with respect to CAD assembly representation in Section 4.7.

4.2 Motivation to seek generative processes

This section presents the benefits of modeling processes to structure a B-Rep component. It shows the limits of CAD construction trees in mechanical design and explains why it is mandatory to set-up generative processes adapted to idealization processes.

[1] generative processes represent ordered sequences of processes emphasizing the shape evolution of the B-Rep representation of a CAD component.

4.2.1 Advantages and limits of present CAD construction tree

As observed in Section 1.2.3, a mechanical part is progressively designed in a CAD software using successive form features. This initial generation of the component shape can be regarded as the task where the component shape structure is generated. Usually, this object structure is described by a binary construction tree containing the elementary features, or primitives, generating the object. This construction tree is very efficient to produce a parameterized model of a CAD object. Effectively, the user can easily update the shape of the object when modifying parameters defined within a user-selected feature and then, re-processing the subset of the construction tree located after this primitive. As illustrated in Section 2.4, Robinson et al. [RAM*06] show that a construction tree with adapted features for FEA can be used to easily generate idealized models.

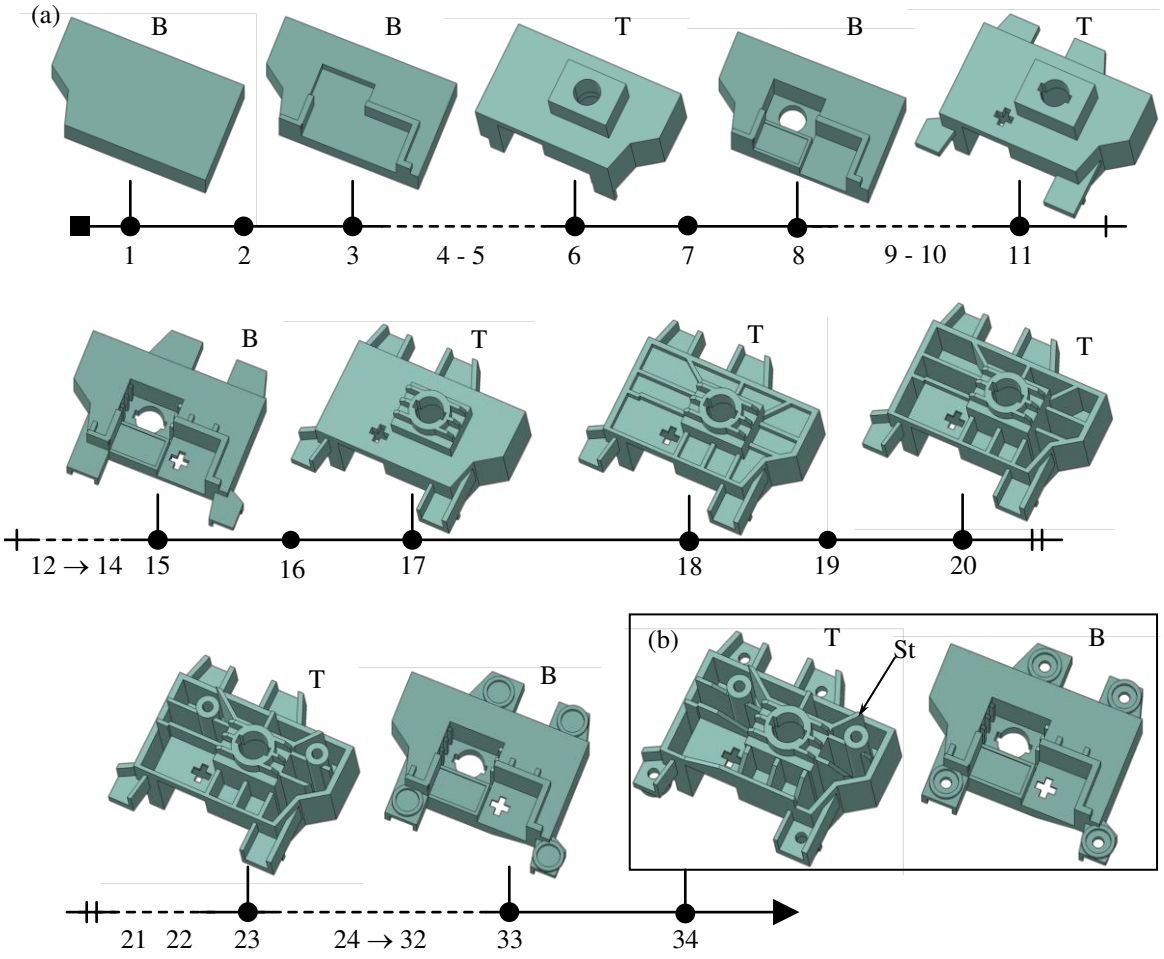


Figure 4.1: An example of a shape generation process: (a) final object obtained after 34 modeling steps and viewed from top (T) and bottom (B), (b) some intermediate shapes obtained after the i^{th} modeling step. The letter T or B appearing with step number indicates whether the object is viewed from top or bottom.

However, the construction tree produced during the design phase may not be suited for the shape decomposition taking place at other stages of a PDP, e.g., during process planning and FEA. Three main issues are preventing the use of current CAD construction trees from FEM pre-processing:

- **The complexity of the final object shape and feature dependencies.** The concept of feature-based design eases the generation of an object shape by adding progressively, and one by one, simple form features. This way, the user starts from a simple solid, i.e., a primitive, and adds or removes volumes using pre-defined features (extrusion, revolution, sweeping, ...) one after the other until he, resp. she, has reached the desired shape of the object. As a consequence of this principle "from simple to complicated", the resulting construction tree can be complex and contains numerous features. Because, the user inserts one form feature at a time, the construction tree is necessarily binary, i.e., each tree node contains one form feature and the object shape obtained after combining this form feature with the object resulting from the previous tree node. As an example, Figure 4.1 illustrates this configuration with a rather complex component where the user's entire modeling process consists of 37 steps, some of them containing multiple contours producing simultaneously several features. Two views, defined as top and bottom in Figure 4.1b, show the major details of the object shape. Figure 4.1a depicts some of the 34 steps involving either extrusion or revolution operations and incorporating either material addition or removal as complementary effects when shaping this object. The parent/child dependencies between form features further increase the complexity of this construction process. The suppression or modification of a parent feature is not always possible due to geometric inconsistencies generated in subsequent tree steps when parent/child dependencies cannot be maintained or when the object boundary cannot produce a solid. This is particularly inconvenient in the context of FEM pre-processing which aims at eliminating detail features to simplify component shapes;

- **The non-uniqueness and user dependence.** Construction trees are not unique, i.e., different users often generate different construction trees for the same final shape. The choice of the sequence of features is made by the designer and depends on his, resp. her, own interpretation of the shape structure of the final object. In current industrial practices, specific modeling rules limit the differences in construction tree generation but they are not dedicated to FEM requirements as explained in Section 3.3;

- **The construction tree availability.** Construction trees contain information which is very specific to each CAD system and each software has its own data structures to represent this construction scheme. Most of the time, this information is lost when transferring objects across CAD systems or even across the

different phases of a PDP. Practically, when using STEP neutral format [ISO94, ISO03], definition of construction tree structures associated with parametric modeling are not preserved. Indeed, to edit and to modify a shape, the parametric relations taking part to the construction tree would need also to be exported. This is difficult to obtain, e.g., even during the upgrade of CATIA software [CAT14] from V4 to V5, the transfer was not fully automatic and some information in construction trees was lost.

As it appears in Figure 4.1, a shape generative process can be rather complex and, even if it is available, there is no straightforward use or transformation of this process to idealize this object (even though its shape contains stiffeners and thin areas that can be modeled with plate or shell elements rather than volume ones). With respect to the idealization objectives, it appears mandatory to set-up another generative process that could incorporate features or primitives with their shapes being close enough to that of stiffeners and thin wall areas.

4.2.2 A new approach to structure a component shape: construction graph generation

Construction trees are important because an object submitted to a FEA preparation process can be subjected to different simplifications at different levels of its construction process. One key issue of these trees is their use of primitives that are available in common industrial CAD software. Another problem lies in the storage of the shape evolution from the initial simple primitive shape to the final object. This principle 'from simple to complicated' matches the objective of using the tree contents for idealization and simplification purposes. Indeed, obtaining simpler shapes through construction processes could already reduce the number of geometric transformations. However, because the CAD construction tree is presently unique, not always available and complicated to modify, its use is difficult. The proposed approach here, structuring the shape of a component, consists in producing generative processes of its B-Rep model that contain sets of volume primitives so that their shapes are convenient for FE simulation. The benefits of extracting new generative processes, as ordered sequences of processes emphasizing the shape evolution of the B-Rep representation of a CAD component, are:

- **To propose a compact shape decomposition adapted to the idealization objectives.** Extraction of compact generative processes aims at reducing their complexity while getting a maximum of information about their intrinsic form features. The proposed geometric structure decomposes an object into volume sub-domains which are independent from each other and close enough to regions that can be idealized. This segmentation method differs from the divide and conquer approaches of [Woo14] because generative processes contain volume prim-

itives having particular geometric properties, e.g., extruded, revolved or swept features. The advantage of these primitives is to ease the validation of the idealization criteria. Indeed, one of their main dimensional characteristics is readily available. For example, the extrusion distance, revolve angle or sweep curve, reduces the primitive analysis to the 2D sketch of the feature. Moreover, interfaces between primitives are identified during the extraction of each generative process to extend the analysis of individual primitives through the whole object and to enable the use of the engineer's FE connection requirements between idealizable regions;

- **To offer the user series of construction processes.** In a CAD system, a feature tree is the unique available definition of the component's construction but is only one among various construction processes possible. Furthermore, in a simulation context, it is difficult to get an adequate representation of the simulation model which is best matching the simulation requirements because the engineer's know-how takes part to the simulation model generation (see Section 3.4). Consequently, the engineer may need to refer to several construction processes to meet the simulation objectives as well as his, resp. her, shape transformation requirements. Providing the engineer with several construction processes helps him, resp. her, generate easily a simulation model. The engineer will be able to navigate shape construction processes and obtain a simpler one within a construction tree that meets the idealization requirements in the best possible way;

- **To produce a shape parameterization independent from any construction tree.** The construction tree is a well-known concept for a user. Parent/child dependencies between features, generated through sketches and their reference planes, ease the interactive design process for the user but creates geometric dependencies difficult to understand for the user. Performing modifications of a complex shape remains difficult to understand for the user, e.g., the cross influence between features located far away in the construction tree is not easy to anticipate. Considering the generation of a construction graph, the proposed approach does not refer to parent/child dependencies between features, i.e., features are geometrically independent each other. The morphological analysis required (see Section 3.4) does not refer to a need for a parameterized representation of components, i.e., each primitive does not need to refer to dependencies between sketch planes. Components can be modified in a CAD software and their new geometry can be processed again to generate a construction graph without referring to the aforementioned dependencies.

As a conclusion, one can state that enabling shape navigation using primitive features similar to that of CAD software is an efficient complement to algorithmic approaches reviewed in Chapter 2 and construction trees. More globally, construction graphs can support both efficiently.

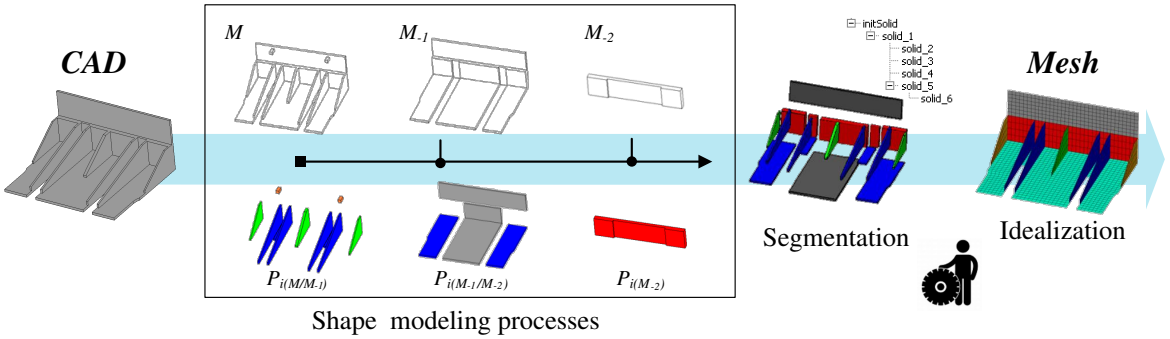


Figure 4.2: An example of shape analysis and generative construction graph.

To this end, this chapter proposes to extract a construction graph from B-Rep CAD models so that the corresponding generative processes are useful for mechanical analysis, particularly when idealization processes are necessary. The graph is extracted using a primitive removal operator that simplifies progressively the object's shape. One could says that the principle is to go 'backward over time'. This characteristic of construction trees is consistent with the objective of simplification and idealization because the shapes obtained after these operations should get simpler. Figure 4.2 illustrates the extraction of a shape modeling process of a CAD component. Primitives P_i are extracted from a sequence of B-Rep objects M_i which become simpler over time. The set of primitives P_i generates a segmented representation of the initial object which is used to derive idealized FE models.

The following sections detail the whole process of extraction of generative processes from B-Rep shapes.

4.3 Shape modeling context and process hypotheses

Before describing the principles of the extraction of generative processes from a B-Rep shape, this section sets the modeling context and the hypotheses of the proposed approach.

4.3.1 Shape modeling context

As a first step, the focus is placed on B-Rep mechanical components being designed using solid modelers. Looking at feature-based modeling functions in industrial CAD systems, they all contain extrusion and revolve operations which are combined with addition or removal of volume domains (see Figure 4.3a). The most common version of the extrusion, as available in all CAD software, is defined with an extrusion direction

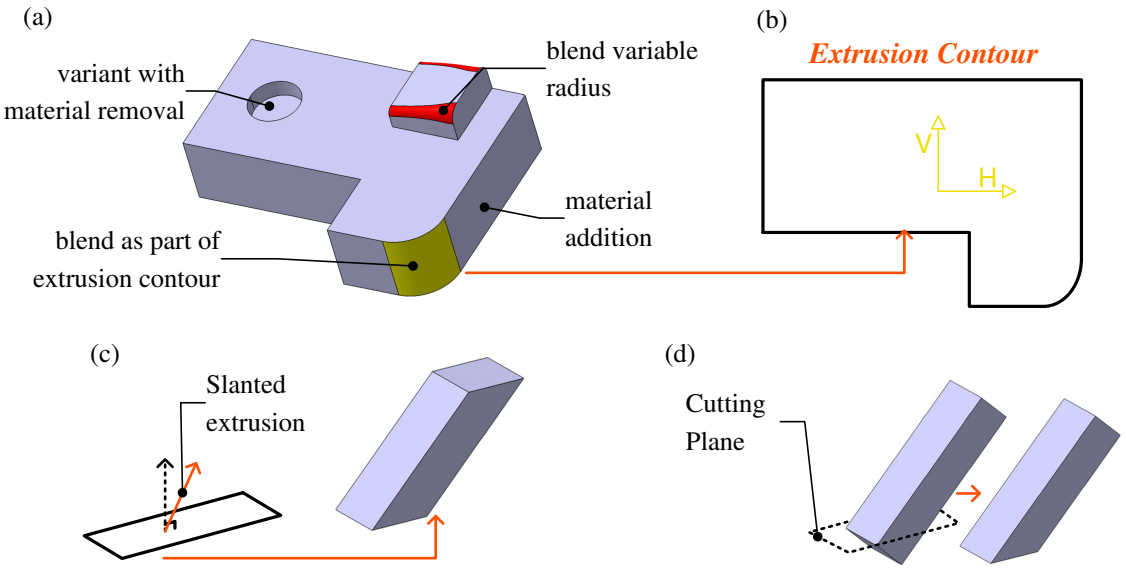


Figure 4.3: (a) Set of basic volume modeling operators, (b) sketch defining an extrusion primitive in (a), (c) higher level volume primitive (slanted extrusion), (d) reference primitive and its first 'cut' transformation to generate the object in (c).

orthogonal to a plane containing the primitive contour. Such an extrusion, as well as the revolution, are defined here as the *reference primitives*. These feature-based B-Rep operations can be seen as equivalent to regularized Boolean operations as available also in common hybrid CAD modelers, i.e., same primitive shapes combined with union or subtraction operators. Modelers also offer other primitives to model solids, e.g., draft surfaces, stiffeners, or free-form surfaces from multiple sections. Even though we don't address these primitives here, it is not a limitation of our method. Indeed, draft surfaces, stiffeners, and similar features can be modeled with a set of *reference primitives* when extending our method to extrusion operations with material removal and revolutions. Appendix B illustrates the simple features and boolean operations available in CAD software and show that it can mainly be reduced to additive/removal extrusion/revolution features in order to cover the present software capabilities . Figure 4.3c illustrates some examples, e.g., an extrusion feature where the extrusion direction is not orthogonal to the sketching plane used for its definition. However, the resulting shape can be decomposed into an extrusion orthogonal to a sketching plane and 'cuts' (see Figure 4.3d) if the generation of a slanted extrusion is not available or not used straightforwardly. Indeed, these construction processes are equivalent with respect to the resulting shape.

Another category of form features available from B-Rep CAD modelers are blending radii. Generally, they have no simple equivalence with extrusions and revolutions. Generated from B-Rep edges, they can be classified into two categories:

1- constant radius blends that can produce cylindrical, toroidal or spherical surfaces;

2- constant radius blends attached to curvilinear edges and variable radius blends.

Category 1 blends include extrusion and revolution primitives and can be incorporated in their corresponding sketch (see Figure 4.3a). This family of objects is part of the current approach. Category 2 blends are not yet addressed and are left for future work. Prior work in this field [VSR02, ZM02, LMTS*05] can be used to derive M from the initial object M_I to be analyzed, possibly with user's interactions.

In summary, all reference primitives considered here are generated from a sketching step in a plane defining at least one closed contour. The contour is composed of line segments and arcs of circles, (see Figure 4.3b). This is a consequence of the previous hypothesis reducing the shapes addressed to closed surfaces bounded by planes, cylinders, cones, spheres, tori, and excluding free-form shapes in the definition of the object boundary. This is not really restrictive for a wide range of mechanical components except for blending radii. The object M to be analyzed for shape decomposition is assumed to be free of blending radii and chamfers that cannot be incorporated into sketched contours. The generative processes are therefore concentrated on extrusion primitives, in a first place, in order to reduce the complexity of the proposed approach. Further hypotheses are stated in the following sections.

4.3.2 Generative process hypotheses

Given a target object M to be analyzed, let us first consider the object independently of the modeling context stated above. M is obtained through a set of primitives combined together by adding or removing material. Combinations of primitives thus create interactions between their bounding surfaces, which, in turn, produce intersection curves that form edges of the B-Rep M. Consequently, edges of M contain traces of generative processes that produced its primitives. Hence, following Leyton's approach [Ley01], these edges can be seen as memory of generation processes where primitives are sequentially combined.

Current CAD modelers are based on strictly sequential processes because the user can hardly generate simultaneous primitives without looking at intermediate results to see how they combine/interact together. Consequently, B-Rep operators in CAD modelers are only binary operators and, during a design process, the user-selected one combines the latest primitive generated to the existing shape of M at the stage t of this generative process. Additionally, CAD modelers providing regularized Boolean operators reduce them to binary operators, even though they are n-ary ones, as classically defined in the CSG approaches [Man88]. Here, the proposed approach does not make any restriction on the amount of primitives possibly generated 'in parallel', i.e., the arity of the combination operators is $n \geq 2$. The generated processes benefit from this hypothesis by compacting the construction trees nodes. This property is illustrated in the result Section 4.6 of this chapter in Figure 4.22.

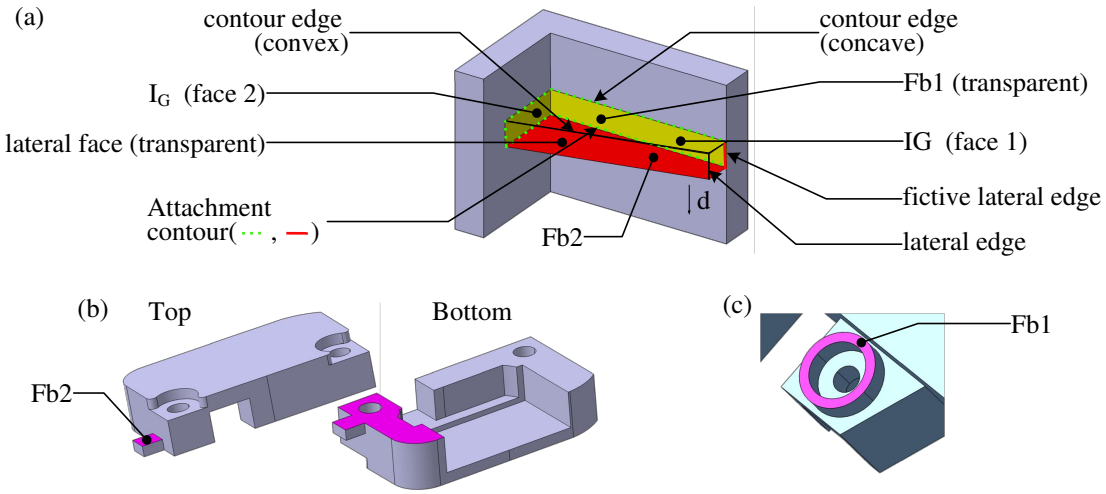


Figure 4.4: a) Entities involved in an extrusion primitive. Visible extrusion feature with its two identical base faces Fb_1 and Fb_2. (b) Visible extrusion feature with its two different base faces Fb_1 and Fb_2. (c) Visible extrusion feature with a unique base face Fb_1 (detail of Figure 4.1a - 34B).

Hypothesis 1: Maximal primitives

The number of possible generative processes producing M can be arbitrary large, e.g., even a cube can be obtained from an arbitrary large number of extrusions of arbitrary small extent combined together with a union operator. Therefore, the concept of maximal primitives is introduced so that the number of primitives is finite and as small as possible for generating M.

> A valid primitive P_i identified at a stage t using a base face F_{b_1} is said to be *maximal* when no other valid primitive P_j at that stage having F'_{b_1} as base face can be entirely inserted in P_i (see Section 4.4.2 and Figure 4.4a): $\forall P_j, P_j \not\subset P_i$. F_{b_1} is a maximal face as defined at Section 4.3.3.

Maximal primitives imply that the contour of a sketch can be arbitrary complex, which is not the case in current engineering practice, where the use of simple primitives eases the interactive modeling process, the parameterization, and geometric constraint assignments to contours. The concept of maximal primitive is analog to the concept of maximal volume used in [WS02, Woo03, Woo14]. Yet, this concept is no used in feature recognition techniques [JG00]. Even if making use of maximal primitives considerably reduces the number of possible generative processes, they are far from being unique for M.

Hypothesis 2: Additive processes

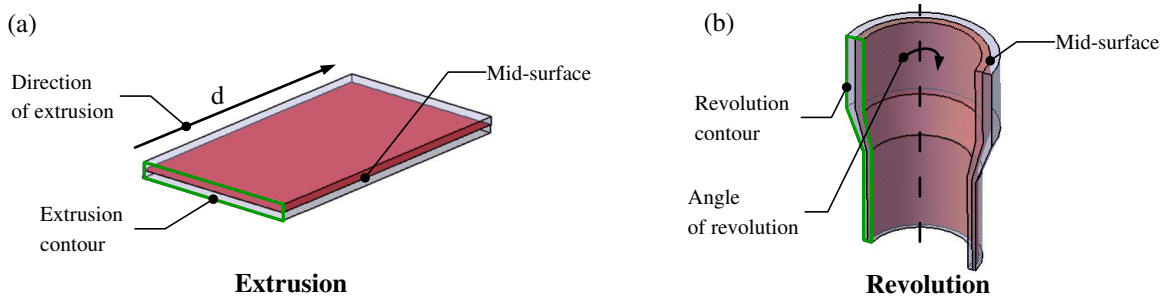


Figure 4.5: Illustrations of two additive primitives: (a) an extrusion primitive and (b) a revolution one. The mid-surfaces of both primitives lie inside their respective volumes.

We therefore make the further hypothesis that the generative processes we are looking for are principally of type additive, i.e., they are purely based on a regularized Boolean union operator when combining primitives at each stage t of generative modeling processes. This hypothesis is particularly advantageous when intending to tailor a set of generative processes that best fit the needs of idealization processes. Indeed, idealized structures, such as mid-surfaces, lie inside such primitives, and connections between primitives locate also the connections between their idealized representatives. Figure 4.5 illustrates an extrusion and a revolution primitives. With both of them, the 3D solid of the primitive includes its mid-surface. Therefore, the idealized representation of M can be essentially derived from each P_i and its connections, independently of the other primitives in case of additive combinations. Figure 4.6 gives an example where M can be decomposed into two primitives combined with a union (b). M in Figure 4.6, (b) can thus be idealized directly from these two primitives and their interface. On the contrary, when allowing material removal, idealization transformations are more complex to process, while the resulting volume shapes are identical. Figure 4.6c shows two primitives which, combined by Boolean subtraction, result also in object (a). However, computing an idealization of (a) by combining idealizations of its primitives in (c) is not possible.

Performing the idealization of M from its primitives strengthens this process compared to previous work on idealization [CSKL04, Rez96, SSM*10] of solids presented in Section 2.3.1 for two reasons. Firstly, each P_i and its connections bound the 3D location of and the connections with other idealized primitives. Secondly, different categories of connections can be defined, which is important because idealization processes still rely on the user's know-how to process connections significantly differing from reference ones. The next Chapter 5 explains in details how to connect mid-surfaces using a taxonomy of connections between extrusion primitives.

Hypothesis 3: Non trivial variants of generative processes

To further reduce the number of possible generative processes, the processes described should be non trivial variants of processes already identified. For example, the

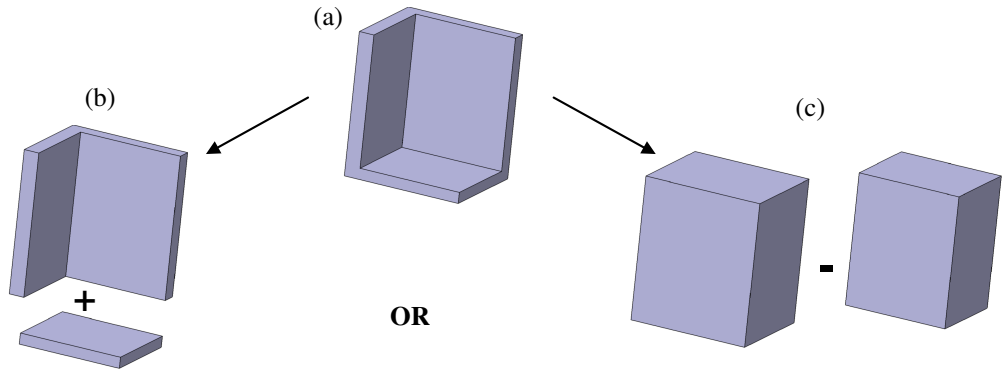


Figure 4.6: a) Simple shape with idealizable sub-domains, (b) Primitives to obtain (a) with an additive process, (c) Primitives to obtain (a) with a removal process.

same rectangular block can be extruded with three different face contours and directions but they create the same volume. Two primitives generating the same shape are considered as the same non-trivial primitive. If the resulting shape of two processes at the j^{th}-level of a construction is the same then, these two processes are said equivalent and are reduced to a single one and the object shape at the $(j-1)^{th}$-level is also unique. These equivalent processes can be detected when comparing the geometric properties of the contours generating this same shape. Other similar observations will be addressed in the following sections when describing the criteria to select meaningful generative processes.

The above hypotheses aim at reducing the number of generative processes producing the same object M while containing primitives suited to idealization transformations, independently of the design process initially set up by engineers.

Conclusion

The overall approach can be synthesized through the process flow of Figure 4.7. The input STEP file contains the B-Rep model M. A set of generative processes is extracted that form sets of construction trees, possibly producing a graph. To this end, application dependent criteria are used to identify one or more construction trees depending on the application needs. Here, we focus on criteria related to idealization for FEA.

4.3.3 Intrinsic boundary decomposition using maximal entities

In order to extract generative processes from the B-Rep decomposition of M, it is important to have a decomposition of M, i.e., topology description, that is intrinsic to

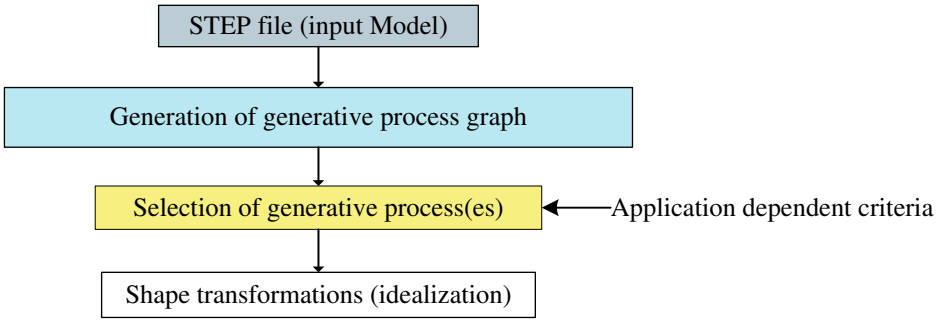


Figure 4.7: Pipeline producing and exploiting generative shape processes.

its shape. A B-Rep decomposition of an object is however not unique (and thus not suitable), because it is subjected to two influences:

- Its modeling process, whether it is addressed forward during a design process or backward as in the present work. Indeed, each operation involving a primitive splits/joins boundary faces and edges of the solid. When joining adjacent faces or edges, their corresponding surfaces or curves can be identical. Their decomposition is thus not unique. However, CAD modelers may not merge the corresponding entities, thus producing a boundary decomposition that is not changing the object shape (see Figure 4.8a). For the proposed approach purposes, such configurations of faces and edges must lead to a merging process so that the object boundary decomposition is unique for a given shape, i.e., it is intrinsic to the object shape;

- The necessary topological properties to setup a consistent paving of an object boundary, i.e., the boundary decomposition must be a CW-complex. Consequently, curved surfaces need to be partitioned. As an example, a cylinder is decomposed into two half cylinders in most CAD modelers or is described with a self connected patch sewed along a generatrix (see Figure 4.8b). In either case, the edge(s) connecting the cylindrical patches are adjacent to the same cylindrical surface and are not meaningful from a shape point of view. Hence, for the proposed approach purposes, they must not participate to the intrinsic boundary decomposition of the object.

Following these observations, the concepts of *maximal faces* and *edges* introduced by [FCF*08] is used here as a means to produce an intrinsic and unique boundary decomposition for a given object M. Maximal faces are identified first. For each face of M, a maximal face F is obtained by repeatedly merging an adjacent face F_a sharing a common edge with F when F_a is a surface of same type and same parameters than F, i.e., same underlying surface. F is maximal when no more face F_a can be merged with F. Indeed, maximal faces coincide with 'c-faces' defined in [Sil81] that have been proved to uniquely defined M. Similarly, for each edge of M, a maximal edge E with

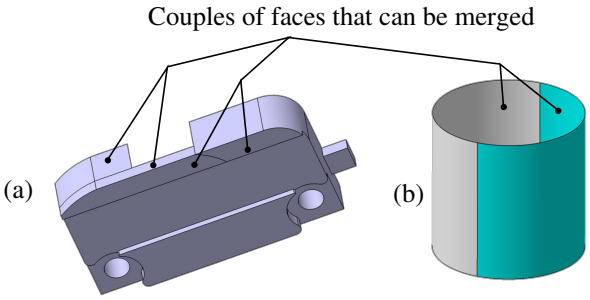


Figure 4.8: Examples of configurations where faces must be merged to produce a shape-intrinsic boundary decomposition: (a) face decomposition due to the modeling process, (b) face decomposition due to topological requirements.

adjacent faces F_1 and F_2 is obtained by repeatedly merging an adjacent edge E_a when E_a is also adjacent to F_1 and F_2. Again, E is maximal when no more edge E_a can be merged with E. As a consequence of these merging processes, it is possible to end up with closed edges having no vertex or with closed faces having no edge. An example for the first case is obtained when generating the maximal face of the cylinder in Figure 4.8b. A sphere described with a single face without any edge and vertex is an example for the second case.

Because of maximal edges without vertices and faces without edges, merging operations are performed topologically only, i.e., the object's B-Rep representation is left unchanged. Maximal faces and edges are generated not only for the initial model M but also after the removal of each primitive when identifying the graph of generative processes. Consequently, maximal primitives (see Hypothesis 1) are based on maximal faces and edges even if not explicitly mentioned throughout this document. Using the concept of *maximal faces* and *edges* the final object decomposition is independent of the sequence of modeling operators.

4.4 Generative processes

Having define the modeling hypotheses and context in the previous Section 4.3, this section presents the principles of the construction of generative processes from B-Rep object. It explains how the primitives are identified and how to remove them from an object M.

4.4.1 Overall principle to obtain generative processes

Preliminary phase

As stated in Section 4.3.1, a preliminary step of the method is to transform it into a blending radii-free object M. To this end, defeaturing functions available in most CAD

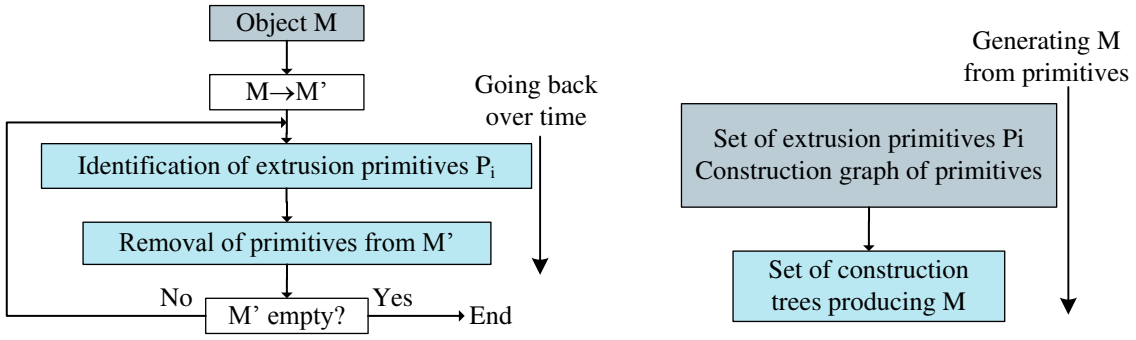


Figure 4.9: Overall scheme to obtain generative processes.

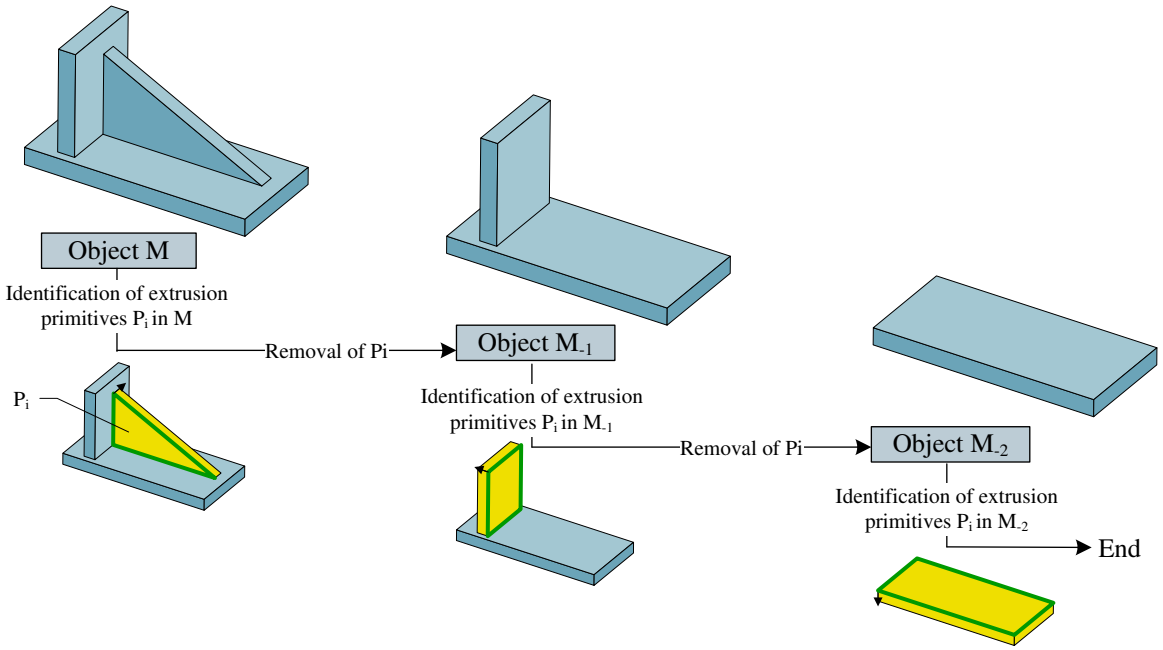


Figure 4.10: An example illustrating the successive identification and removal of primitives.

systems are applied. This operation is a consequence of the modeling context defined in Section 4.3.1. Even though these functions may not be sufficient and robust enough, this is the current working configuration. In contrast to blending radii, most chamfers are included in the present approach because they can be part of extrusion primitives and hence, included in the sketched contours used to define extrusion primitives. Even if CAD software provide specific functions for chamfers, they are devoted to the design context but basic operators of extrusion with material addition or removal could produce the same result, in general. This analysis regarding chamfers shows the effect of the concept of maximal primitives (see Hypothesis 1).

Main phase

Starting with the solid M, the generative processes are obtained through two phases:

- M is processed by iterative identification and removal of primitives. The objective of this phase is to 'go back in time' until reaching root primitives for generative processes. The result of this phase is a set of primitives;

- Based on hypotheses of Section 4.3.2, a set of generative processes is produced using the primitives obtained at the end of the first phase to meet the requirements of an application: here idealization (see Chapter 5).

Finally, the decomposition $\mathcal{D}$ of M into extrusion primitives is not limited to a single construction tree but it produces a construction graph G_D iteratively generated from M. G_D contains all possible non trivial construction trees of M (see Hypothesis 3). The process termination holds whenever M is effectively decomposable into a set of extrusion primitives. Otherwise, $\mathcal{D}$ is only partial and its termination produces either one or a set of volume partitions describing the most simplest objects $\mathcal{D}$ can reach.

Figure 4.9 summarizes the overall scheme just described previously. When generating G_D, we refer to $M = M_0$ and evolutions M_{-j} of it backward at the j^{th} step of $\mathcal{D}$. Figure 4.10 illustrates the major steps of the extraction of a generative process graph, i.e., from the primitive identification up to its removal from M, and will be further explained in Sections 4.4.2 and 4.4.3.

4.4.2 Extrusion primitives, visibility and attachment

In order to identify extrusion primitives P_i in $M = M_0$ and evolution M_{-j} of it, backward at the j^{th} step of the generation of the generative process graph, it is mandatory to define its geometric parameters as well as the hypotheses taken in the present work (see Figure 4.4).

First of all, let us notice that a 'reference primitive' P_i is never appearing entirely in M or M_{-j} unless it is isolated like a root of a construction tree, i.e., $P_i = M$ or $P_i = M_{-j}$. Apart from these particular cases, P_i are only partly visible, i.e., not all faces of P_i are exactly matching faces of M_{-j}. For simplicity, we refer to such P_i as 'visible primitives'. P_i is the memory of a generative process that took place between M_{-j} and $M_{-(j+1)}$. Extracting P_i significantly differs compared to feature recognition approaches [Rez96, LGT01, WS02, SSM*10, JG00]. In feature recognition approaches, P_i is identified through validity constraints with its neighboring attachment in M, i.e., faces and edges around P_i. These constraints limits the number of possible primitives by looking to the best interpretation of some visible boundaries of the object M. Here, identifying visible primitives enables the generation of reference ones having simpler contours. Only the visible part of the primitive is used to identify the primitive in M, without restricting the primitive to the visible boundaries of M. The proposed identification process of P_i is more general, it does not integrate any validity constraint on

the attachment of P_i with M. This constraint released, this process enables the identification of a greater number of primitives which can be compared with each other not only through their attachment to M but also through their intrinsic shape complexity.

Definition of the primitive

The parameters involved in a reference extrusion P_i are the two base faces, Fb_1 and Fb_2, that are planar and contain the same sketched contour where the extrusion takes place. Considering extrusions that add volume to a pre-existing object, the edges of Fb_i are called *contour edges* which are all convex. Indeed, P_i being standalone primitive, all its contour edges are convex. A convex edge is such that the outward normals of its adjacent faces define an angle α where: $0 < \alpha < \pi$. When P_i belongs to M_{-j}, the contour edges along which P_i is attached to M_{-j} can be either convex or concave depending on the neighborhood of P_i in M_{-j} (see Figure 4.4a).

In the direction $\mathbf{d}$ of the extrusion, all the edges are straight line segments parallel to each other and orthogonal to Fb_i. These edges are named *lateral edges*. Faces adjacent to Fb_i are called *lateral faces*. They are bounded by four edges, two of them being lateral edges. Lateral edges can be *fictive lateral edges* when a lateral face coincides with a face of M_{-j} adjacent to P_i (see Figure 4.4a). When lateral faces of P_i coincide with adjacent faces in M_{-j}, there cannot be edges separating P_i from $M_{-(j+1)}$ because of the definition of maximal faces. Such a configuration refers to *fictive base edges* (see Figure 4.11 with the definition of primitive P_1).

Principle of primitive identification: Visibility

The *visibility* of P_i depends on its insertion in M_{-j} and sets the conditions to identify P_i in ∂M_{-j}[2]. An extrusion primitive P_i can be *visible* in different ways depending on its insertion in a current object M_{-j}. The simplest visibility is obtained when P_i's base faces Fb_i in M_{-j} exist and when at least one lateral edge connects Fb_i in M_{-j} (see Figure 4.4a and 4.11(step1)).

More generally, the contour of Fb_1 and Fb_2 may differ from each other (see Figure 4.4b) or the primitive may have only one base face Fb_1 visible in M_{-j} together with one existing lateral edge that defines the minimal extrusion distance of Fb_1 (see Figure 4.4c). Our two hypotheses on extrusion visibility thus state as follows:

- First, at least one base face Fb_i is visible in M_{-j}, i.e., the contour of either Fb_1 or Fb_2 coincides with a subset of the attachment contour of P_i in M_{-j};
- Second, one lateral edge exists that connects Fb_i in M_{-j}. This edge is shared by two lateral faces and one of its extreme vertices is shared by Fb_i.

[2] ∂M_{-j} is the boundary of the volume object M, i.e., the B-Rep representation

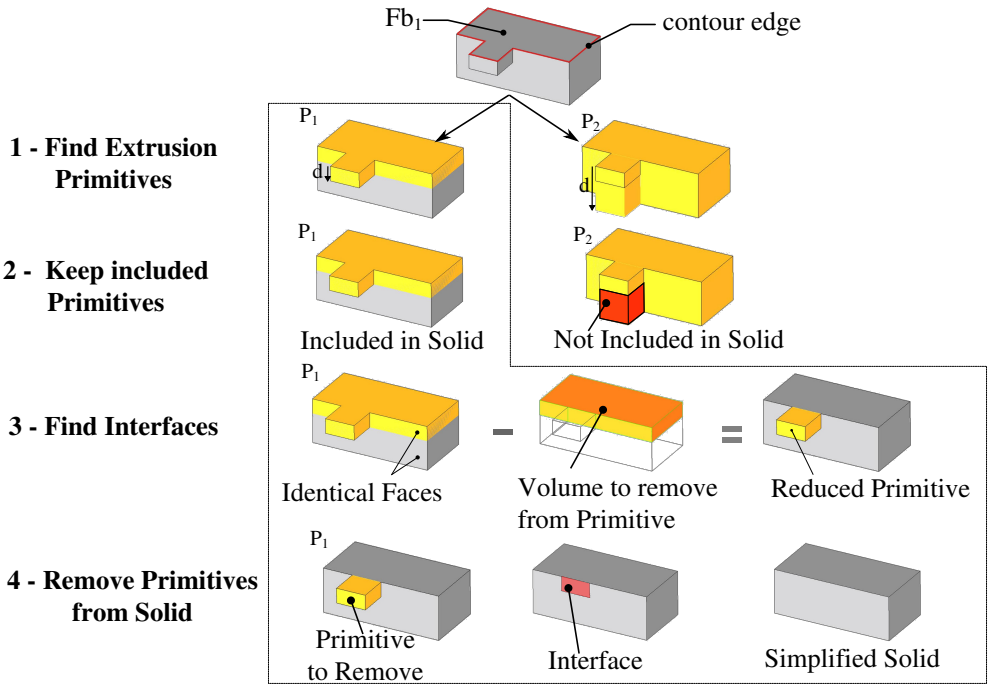


Figure 4.11: An example illustrating the major steps to identify a primitive P_i and remove it from the current model M_{-j}.

P_i is entirely defined by Fb_i and the extrusion length obtained the maximum length of the generatrix of P_i extracted from its lateral faces partly or entirely visible in M_{-j}. Notice that the lateral edges mentioned may not be maximal edges when lateral faces are cylindrical because maximal faces may remove all B-Rep edges along a cylindrical area. These conditions of definition of extrusion distance restricts the range of extrusion primitives addressed compared to the use of the longest lateral segment existing in the lateral faces attached to Fb_i. However, it is a first step enabling to address a fair set of mechanical components and validate the major concepts of the proposed approach. This generalization is left for future work. Figure 4.4b, c give examples involving two or one visible base faces, respectively.

Attachment

An extrusion primitive P_i is *attached* to M_{-j} in accordance to its visibility in M_{-j}. The attachment defines a geometric interface, I_G, between P_i and $M_{-(j+1)}$, i.e., $I_G = P_i \cap M_{-(j+1)}$. This interface can be a **surface or a volume or both**, i.e., a non-manifold model. One of the simplest attachments occurs when P_i has its base faces Fb_1 and Fb_2 visible. This means that P_i is connected to $M_{-(j+1)}$ through lateral faces only. Consequently, I_G is a surface defined by the set of lateral faces not visible in P_i. Figure 4.4a illustrates such a type of interface (I_G contains two faces depicted in yellow).

Simple examples of attachment I_G between P_i and $M_{-(j+1)}$ are given in Figure 4.4.

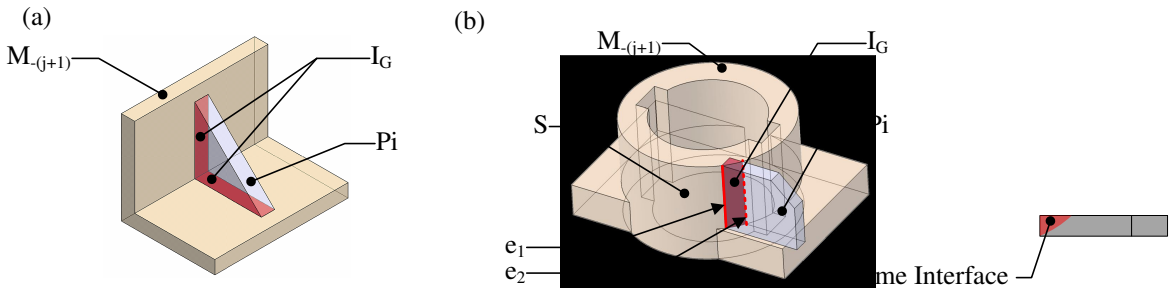


Figure 4.12: Example of geometric interface I_G between P_i and $M_{-(j+1)}$: (a) surface type, (b) volume type.

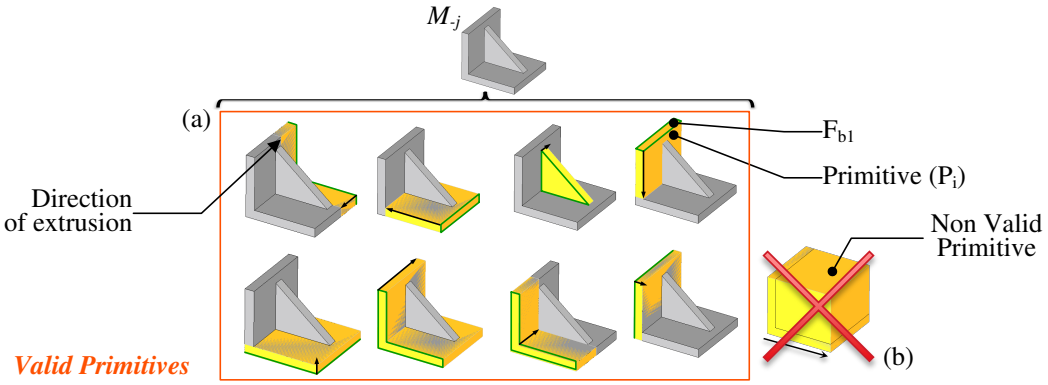


Figure 4.13: Collection of primitives identified from M_{-j}: (a) Valid primitives included in M_{-j}, (b) invalid primitive because it is not fully included in M_{-j}. Green edges identify the contour of the base face of the primitive.

4.4a involves a surface interface and 4.4b illustrates a volume one. Let us notice that the interface between P_i and $M_{-(j+1)}$ in 4.4b contains also a surface interface located at the bottom of the primitive that is not highlighted. However, as we will see in Section 4.5, all possible variants of I_G must be evaluated to process the acceptable ones.

In a first step, P_i can be translated directly into an algorithm to identify them (procedure $find\_visible\_extrusion$ of algorithm 1). The visibility of P_i does not refer to its neighboring faces in M_{-j}. Next, they are subjected to validity conditions described in the following section.

4.4.3 Primitive removal operator to go back in time

The purpose is now to describe the removal operator that produces a new model $M_{-(j+1)}$ anterior to M_{-j}. This removal operator is defined as a binary operator with P_i and M_{-j} as operands and $M_{-(j+1)}$ as result. In the context of a generative process, M_{-j} relates to a step j and $M_{-(j+1)}$ to a step $(j+1)$.

Characterization of interfaces

In order to be able to generate $M_{-(j+1)}$ once P_i is identified, it is necessary to reconstruct faces adjacent to P_i in M_{-j} so that $M_{-(j+1)}$ defines a volume. To this end, the faces of M_{-j} adjacent to P_i and I_G must be characterized. Here, P_i is considered to be adjacent to other subsets of primitives through one edge at least. The removal operator depends on the type of I_G. Due to manifold property of M, two main categories of interfaces have been identified:

1- I_G is of surface type. In this category, the removal operator will have to create lateral faces and/or the extension of Fb_2 so that the extended face coincides with Fb_1. Indeed, this category needs to be subdivided into two sub categories:

   a- I_G contains lateral faces of P_i only (see Figure 4.4a) or I_G contains also an extension of Fb_2 and edges of this extension are concave edges in $M_{-(j+1)}$;

   b- I_G may contains lateral faces of P_i but it contains an extension of Fb_2 and the edges of this extension are fictive base edges in M_{-j}. These edges would be convex edges in $M_{-(j+1)}$, (see P_1 in Figure 4.11);

2- I_G contains at least one volume sub-domain.

In addition, considering that Fb_1 at least is visible and P_i is also visible (see Section 4.4.2), the attachment contour may not be entirely available to form one or more edge loops (see Figure 4.4a). Also, I_G can contain more than one connected component when P_i is resembling a handle connected to $M_{-(j+1)}$, which produces more than one edge loop to describe the attachment of P_i to $M_{-(j+1)}$ in I_G.

Validity

Whatever the category of interface, once P_i is identified and its parameters are set (contour and extrusion distance), it is necessary to validate it prior to define its interface (step 2 of Figure 4.11). Let P_i designates the volume of the reference primitive, i.e., the entire extrusion P_i. To ensure that P_i is indeed a primitive of M_{-j}, the necessary condition is formally expressed using regularized Boolean operators between these two volumes:

$$(M_{-j} \cup^* P_i) -^* M_{-j} = \phi. \tag{4.1}$$

This equation states that P_i intersects M_{-j} only along the edge loops forming its attachment to $M_{-(j+1)}$, i.e., P_i does not cross the boundary of M_{-j} at other location than its attachment. The regularized Boolean subtraction states that limit configurations producing common points, curve segments or surface areas between P_i and M_{-j} at other locations than the attachment of P_i are acceptable. This condition strongly reduces the number of primitives over time. Figure 4.13 illustrates the list of 9 primitives identified from an object M_{-j}. 8 primitives in 4.13a satisfy the validity criterion as they are included in M_{-j}. The primitive in 4.13b is not fully included in M_{-j} and is

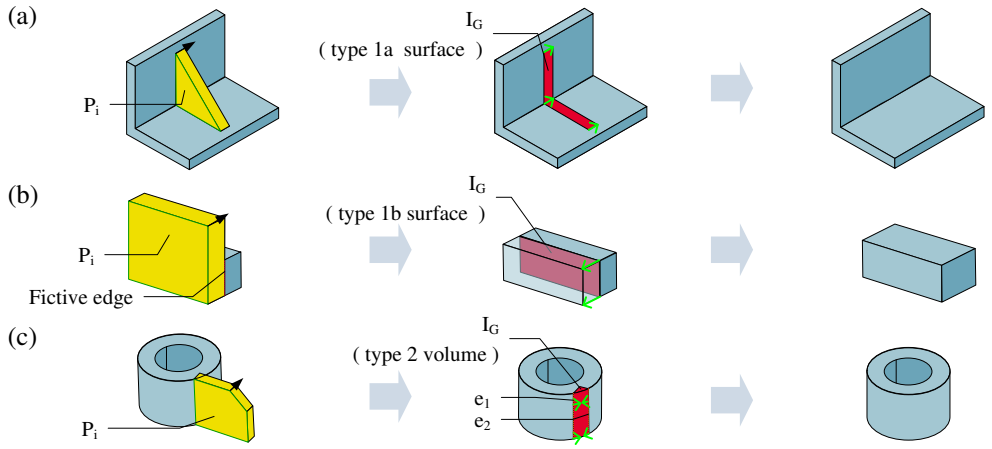


Figure 4.14: Illustration of the removal of P_i with three different interface types: (a) type 1a, (b) type 1b, (c) type 2.

removed from the set. Another example in Figure 4.11 at step 2 shows that primitives P_2 and P_3 can be discarded.

Removal of P_i

The next step is the generation of $M_{-(j+1)}$ once P_i has been identified and removed from M_{-j}. Depending of the type of I_G, some faces of P_i may be added to ensure that $M_{-(j+1)}$ is a volume (see Figure 4.11 steps 3 and 4). For each category of interface between P_i and M_{-j}, the removal operation is described as follow:

- *Type 1a:* If I_G is of type 1a, then the faces adjacent to the contour edges of Fb_1 are orthogonal to Fb_1. These faces are either planar or cylindrical. I_G contains the faces extending these faces, F_{a_1}, to form the lateral faces of P_i that were 'hidden in M_{-j}'. Edges of the attachment of P_i belonging to lateral faces of P_i can be lateral edges (either real or fictive ones) or arbitrary ones. Lateral edges bound faces in F_{a_1}, arbitrary edges bound the extension of the partly visible lateral faces of P_i, they belong to: F_{a_2}. Then, I_G may contain the extension of Fb_2 called F_{a_3} such that: $Fb_2 \cup F_{a_3} = Fb_1$. Then:

$$\partial M_{-(j+1)} = (\partial M_{-j} - \partial P_i) \cup (F_{a_1} \cup F_{a_2} \cup F_{a_3}), \tag{4.2}$$

  where ∂M_{-j} is the set of connected faces bounding M_{-j}, ∂P_i is the set of connected faces bounding the visible part of P_i. $\partial M_{-(j+1)}$ defines a closed, orientable surface, without self intersection. $M_{-(j+1)}$ is therefore a volume. Figure 4.14 a and Figure 4.15 illustrate this process for interface of type 1a;

- *Type 1b:* If I_G is of type 1b, I_G contains a set of faces extending lateral faces of P_i: F_{a_1}. To reduce the description of the various configurations, let us focus on the key aspect related to the extension of Fb_2 contained in I_G. If this extension

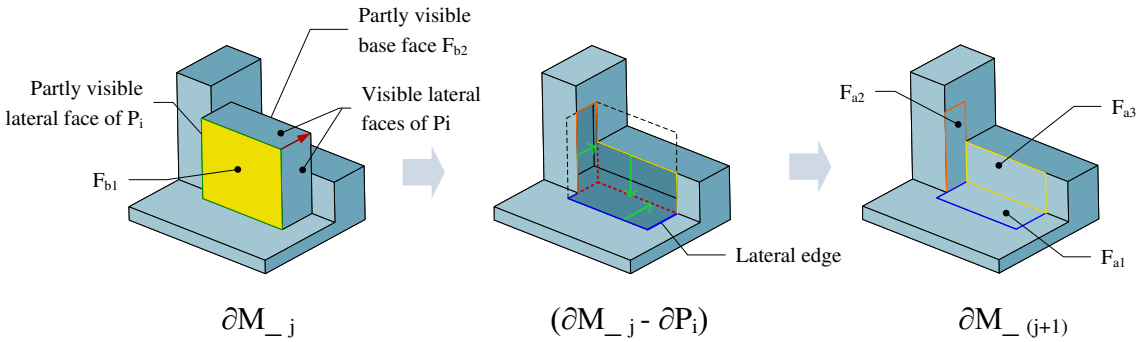


Figure 4.15: Illustration of the removal of P_i for interface of surface type 1a and generation of $\partial M_{-(j+1)}$ with the extension of lateral and base faces.

can be defined like F_{a_3} above, it has to be observed that fictive edges of this extension in M_{-j} are replaced by convex edges in $M_{-(j+1)}$, i.e., edges of the same type (convex) as their corresponding edges in Fb_1 (see Figure 4.11 step 3 left image). Without going into details, these fictive edges can be removed to simplify the contour of P_i since they bring unnecessary complexity to P_i and does not affect the complexity of $M_{-(j+1)}$. In addition to simplify progressively the object's shape, reducing the complexity of primitives' contours is a way to obtain primitives having a form as simple as possible. The corresponding effect is illustrated on Figure 4.11 steps 3 and 4 and on Figure 4.14b. This contour simplification can influence the contents of the sets F_{a_1} and F_{a_3} above but it has no impact on the integrity of the volume $M_{-(j+1)}$ obtained;

- *Type 2:* If I_G belongs to category 2, it contains at least one volume sub-domain. Here again the diversity of configurations can be rather large and it is not intended to give a detailed description of this category. A first condition to generate a volume interface relates to surfaces adjacent to P_i. If S is the extension of such a surface and $S \cap^* P_i \neq \phi$, S may contribute to the generation of a volume sub-domain. Then, each of these surfaces has to be processed. To this end, all the edges attaching P_i in $M_{-(j+1)}$ and bounding the same surface in $M_{-(j+1)}$ are grouped together since they form a subset of the contour of faces possibly contributing to a volume sub-domain. These groups are named E_a. Such an example of edge grouping is given in Figure 4.14b where e_1 and e_2 are grouped because of their adjacency between P_i and the same cylindrical surface. E_a, together with other sets of edges are used to identify loops in S that define a volume sub-domain of I_G that must satisfy validity conditions not described here for sake of conciseness. Figure 4.16 illustrates the identification of a volume interface, S divides P_i into two volume sub-domains and generates a volume interface.

There may be several valid volume sub-domains defining alternative sets of faces to replace the visible part of P_i, ∂P_i, in ∂M_{-j} by sets of faces that promote either the

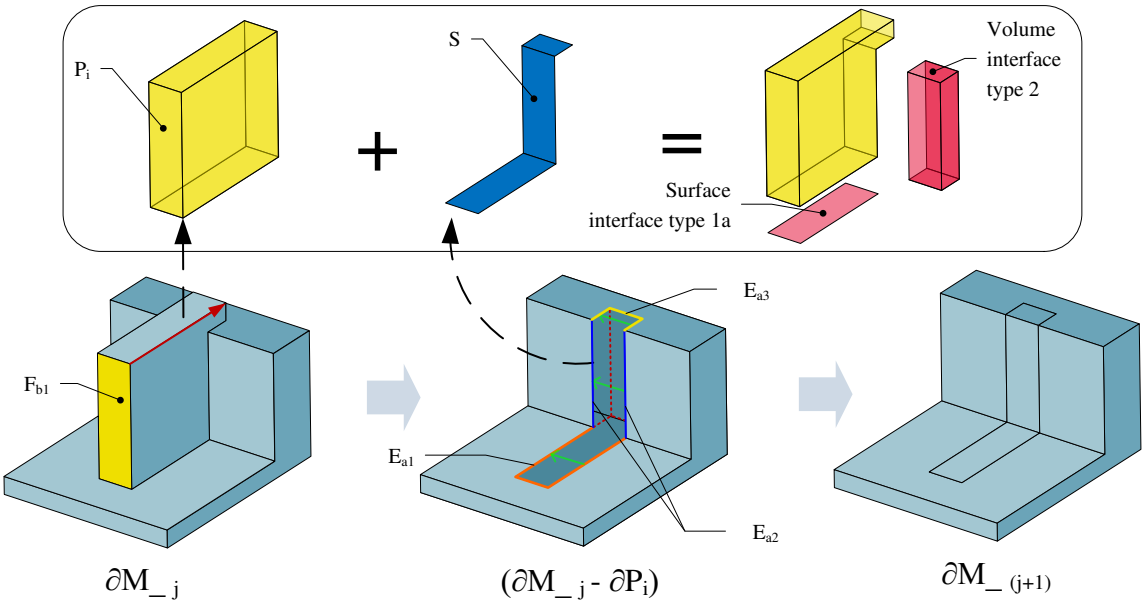


Figure 4.16: Illustration of the removal of P_i containing a volume interface of type 2.

extension of surfaces adjacent to P_i or the imprint of P_i in $M_{-(j+1)}$ with the use of faces belonging to the hidden part of P_i in M_{-j}. All the variants are processed to evaluate their possible contribution to the generative process graph.

If, in a general setting, there may be several variants of I_G to define $M_{-(j+1)}$, these variants always produce a realizable volume, which differs from the half-space decomposition approaches studied in [SV93, BC04] where complement to the half-spaces derived from their initial boundary were needed to produce a realizable volume.

4.5 Extracting the generative process graph

Having defined the primitive removal operator, the purpose is now to incorporate constraints on variants of I_G so that a meaningful set of models M_{-j}, $j > 0$, can be generated to produce a generative process graph.

4.5.1 Filtering out the generative processes

As mentioned earlier, the principle of the proposed approach is to 'go back in time' from model M to single primitives forming the roots of possible construction trees. The main process to select primitives to be removed from M_{-j} is based on a simplification criterion.

Primitive selection based on a shape simplicity concept

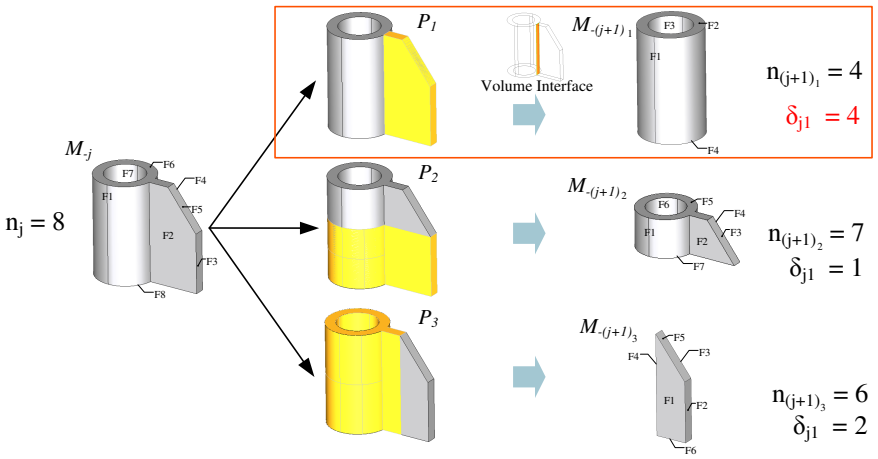


Figure 4.17: Illustration of the simplicity concept to filtering out the generative processes. The number of maximal faces of is greatly reduced with P_1 than P_2 and P_3.

Any acceptable primitive removal at step j of the graph generation must produce a transformation of M_{-j} into k objects $M_{-(j+1)_k}$ using I_{G_k}, one of the variants of I_G, such that $M_{-(j+1)_k}$ is simpler than M_{-j}. This simplicity concept is a necessary condition for the graph generation to converge toward a set of construction trees having a single primitive as root. Consequently, the simplicity concept applied to the transition between M_{-j} and $M_{-(j+1)_k}$ is sufficient to ensure the convergence of the graph generation process.

The shape simplification occurring between M_{-j} and $M_{-(j+1)_k}$ can be defined as follows. First of all, it has to be considered that ∂M_{-j} and $\partial M_{-(j+1)_k}$ contain maximal faces and edges. In fact, after P_i is removed and replaced by I_{G_k} to produce $M_{-(j+1)_k}$, its boundary decomposition is re-evaluated to contain maximal faces and edges only. Then, let n_j be the number of (maximal) faces in M_{-j} and $n_{(j+1)_k}$ be the same quantity for $M_{-(j+1)_k}$, the quantity δ_{jk}:

$$\delta_{jk} = n_j - n_{(j+1)_k} \tag{4.3}$$

characterizes the shape simplification under the variant I_{G_k} if:

$$\delta_{jk} \geq 0. \tag{4.4}$$

This condition is justified because it enforces a 'diminishing number of maximal faces over time', which is an intrinsic quantity to each shape.

Figure 4.17 illustrates the simplicity criterion between three primitives P_1, P_2, and P_3 to be removed from a solid M_{-j}. M_{-j}, the initial solid, contains $n_j = 8$ (maximal) faces. When removing P_1 from M_{-j}, the resulting solid $M_{-(j+1)_1}$ contains $n_{(j+1)_1} = 4$ (maximal) faces. Identically, the resulting solids from P_2 and P_3 contains respectively $n_{(j+1)_2} = 7$ and $n_{(j+1)_3} = 6$ (maximal) faces. As a result, the primitive P_1 is selected because the quantity $\delta_{j1} = 4$ is greater than δ_{j2} and δ_{j3}. By removing P_1, the resulting object M_{-j} is simpler than with P_2 or P_3.

4.5.2 Generative process graph algorithm

Having defined the condition to evolve backward in the generative process graph, the graph generation is summarized with algorithm 1.

Algorithm 1 Extract generative process graph

procedure *Extract_graph* ▷ The main procedure to extract generative processes of a solid M
  inputM
  $node\_list \leftarrow root$; $current\_node \leftarrow root$;
  $arc\_list \leftarrow nil$; $current\_arc \leftarrow nil$; $node\_list(0) = M$
  while $size(node\_list) > 0$ **do** ▷ Stop when all solids M_{-j} reach a terminal primitive root
    $current\_node = last\_element\_of\_list(node\_list)$
    $M_{-j} = get\_solid(current\_node)$
    $config\_list = Process\_variant(M_{-j})$
    $compare\_config(get\_all\_config(graph), config\_list)$ ▷ Compare new variants with the existing graph nodes
    for each *config in config_list* **do**
      $M_{-(j+1)} = remove\_primitives(M_{-j}, config)$ ▷ Remove the identified primitives from M_{-j}
      $node = generate\_node(M_{-(j+1)}, config)$
      $add\_node(graph, node)$
      $arc = generate\_arc(node, current\_node)$
      $add\_arc(graph, arc)$
      $append(node\_list, node)$
    end for
    $remove\_element\_from\_list(node\_list, current\_node)$
  end while
end procedure

procedure $config\_list = Process\_variant(M_{-j})$ ▷ Process each variant M_{-j} to go 'backward in time'
  $initialize\_primitive\_list(prim\_list)$
  $ext\_list = find\_extrusion(M_{-j})$
  for each P_i *in* ext_list **do**
    $P_i = simplify\_prim\_contour(P_i, M_{-j})$
    $interf\_list = generate\_geom\_interfaces(P_i, M_{-j})$
    $interf\_list = discard\_complex(interf\_list, P_i, M_{-j})$
    if $size(interf\_list) = 0$ **then**
      $remove\_from\_list(P_i, ext\_list)$;
    end if
    $append(prim\_list, interf\_list(i))$
  end for
  $sort\_primitive(prim\_list)$
  $config\_list = generate\_independent\_ext(prim\_list, M_{-j},)$
end procedure

procedure $ext\_list = find\_extrusion(M_{-j})$ ▷ Find sets of primitives to be removed from M_{-j}
  $ext\_list = find\_visible\_extrusions(M_{-j})$;
  $ext\_list = remove\_ext\_outside\_model(M_{-j}, ext\_list)$; ▷ Reject primitives not totally included in M_{-j}
  $ext\_list = remove\_ext\_included\_ext(ext\_list)$; ▷ Process only maximal primitives
end procedure

The main procedure *Extract_graph* of the algorithm 1 processes the *node_list* containing the current variants of the model at the current step 'backward in time' using the procedure *Process_variant* and compares the new variants to the existing graph nodes using $compare\_config$. If variants are identical, graph nodes are merged, which creates cycles. Then, *Extract_graph* adds a tree structure to a given variant corresponding to the new simpler variants derived from M_{-j}. The graph is completed when there is no more variant to process, i.e., *node_list* is empty. Here, the purpose is to remove (using *remove_primitives*) the largest possible amount of primitives P_i whose interfaces I_{G_k} are not overlapping each other, i.e., $\forall(i, j, k, l), i \neq j$, $I_{G_k} \in P_i$, $I_{G_l} \in P_j$,

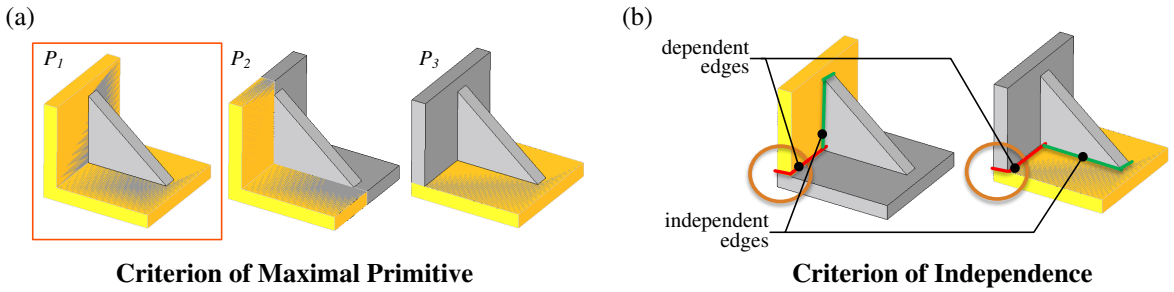


Figure 4.18: Selection of primitives: (a) Maximal primitive criterion not valid for P_2 and P_3 because they are included in P_1, (b) two dependent primitives with common edges in red.

$I_{G_l} \cap I_{G_k} = \phi$, otherwise δ_{jk} would not be meaningful. Selecting the largest possible amount of P_i and assigning them to a graph node is mandatory to produce a compact graph. Each such node expresses the fact that all its P_i could be removed, one by one, in an arbitrary order, which avoids describing trivial ordering changes. The primitive removal operator, described in Section 4.4.3, generates not only simpler solids' shapes but also simplify the primitives' contours (using $simplify\_prim\_contour$). Both simplification effects reduce considerably the complexity of the extracted generative processes compared to the initial construction tree of the CAD component.

To process each variant M_{-j} of M, $Process\_variant$ starts with the identification of valid visible extrusion primitives in M_{-j} using $find\_extrusion$ (see Sections 4.4.2 and 4.4.3 respectively). However, to produce maximal primitives (see Hypothesis 1), all valid primitives which can be included into others (because their contour or their extrusion distance is smaller than the others) are removed ($remove\_ext\_included\_ext$). Figure 4.18a shows two primitives P_2 and P_3 included in a maximal primitive (see Hypothesis 1) P_1.

Once valid maximal primitives (see Hypothesis 1) have been identified, processing the current variant M_{-j} carries on with contour simplification: $simplify\_prim\_contour$, if it does not impact the shape complexity of $M_{-(j+1)}$ (see Section 4.4.3). Then, all the valid geometric interfaces I_{G_k} of each primitive are generated with $generate\_geom\_interfaces$ (see Section 4.4.3) and interfaces I_{G_k} increasing the shape complexity are discarded with $discard\_complex$ to ensure the convergence (see Section 4.5.1). Sets of independent primitives are ordered to ease the user's navigation in the graph. As illustrated in Figure 4.18, two primitives are independent if there is no geometric intersection between them.

4.6 Results of generative process graph extractions

The previous process described in Section 4.5 has been applied to a set of components whose shapes are compatible with extrusion processes to stay consistent with

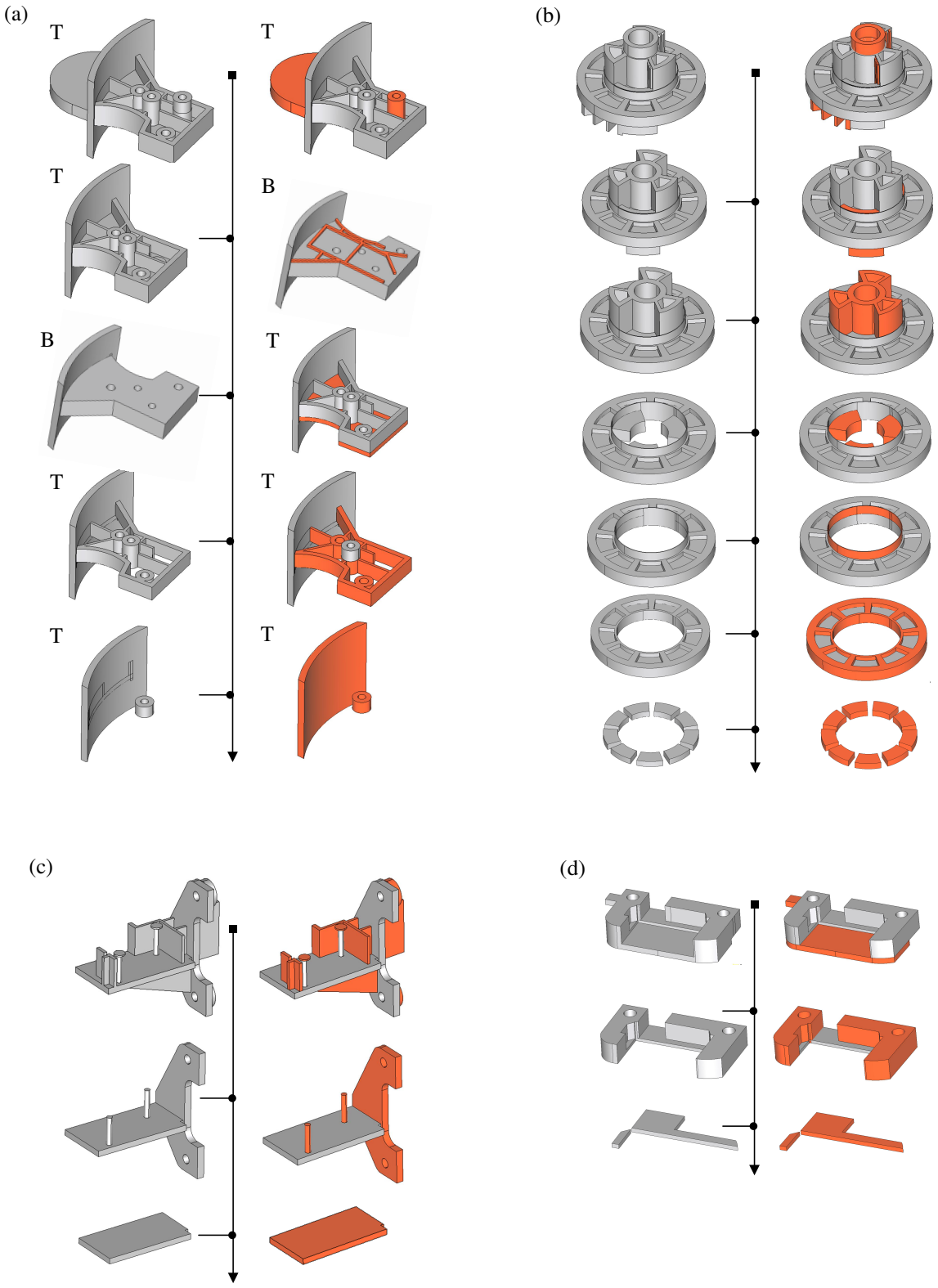


Figure 4.19: Extraction of generative processes for four different components: a, b, c, d. Orange sub-domains highlight the set of visible primitives removed at each step of the graph generation. Construction graph reduces to a tree for each of these components: (a) T and B indicate Top and Bottom views to locate easily the primitives removed. Other components use a single view.

algorithm 1 though they are industrial components. The results have been obtained automatically using algorithm 1 implemented using Python and bindings with Open Cascade (OCC) library [CAS14]. The complexity of Algorithm 1 is $O(n^2)$. Regarding time, most consuming operations refer to the procedure $find\_extrusion$ which uses boolean operations to verify the validity of each primitive. Therefore, the practical performance of the algorithm is dependent on the robustness and complexity of the boolean operators. Statistics given are the amount of calls to a generic Boolean type operator available in the OCC [CAS14] library, the total number of visible primitives ($find\_visible\_extrusions$), n_v, and the final number of P_i in the graph, n_p.

Results on industrial shapes

Figure 4.19 shows the generative processes extracted from four different and rather simple components. They are characterized by triples (n_B; n_v; n_p), (2183; 220; 8), (9353; 240; 31), (8246; 225; 15), (1544; 132; 6), for (a), (b), (c) and (d), respectively. The graph structure reduces to a tree one for each of them. It shows that merging all extrusions in parallel into a single node can be achieved and results into a compact representation. These results also show the need for a constraint that we can formalize as follows: configurations produced by *generate_independent_ext* must be such that each variant $M_{-(j+1)_k}$ generated from M_{-j} must contain a unique connected component as it is with M. However, this has not been implemented yet. This continuity constraint expresses the fact that M is a continuous medium and its design process follows this concept too. Figure 4.21 illustrates this constraint on the construction graph of a simple solid. The object M_{-1_1} is composed of 5 solids which represent a non continuum domain. Consequently, any of its transformation stages must be so to ensure that any simplified model, i.e., any graph node, can stand as basis for an idealization process. Then, it is up to the idealizations and their hypotheses to remove such a constraint, e.g., when replacing a primitive by kinematic boundary conditions to express a rigid body behavior attached to one or more primitives in the graph.

Figure 4.20 shows the graph extracted from the component analyzed in Figure 4.1. It is characterized by (111789; 1440; 62). Two variants appear at step 4 and lead to the same intermediate shape at step 8. It effectively produces a graph structure. It can be observed that the construction histories are easier to understand for a user than the one effectively used to model the object (see Figure 4.1). Clearly, the extrusion primitives better meet the requirements of an idealization process and they are also better suited to dimension modification processes as mentioned in Section 4.1. The current implementation of Algorithm 1 uses high-level operators, e.g., boolean operations, rather than dedicated ones. This implementation limits the time reduction which could be achieved compared to the interactive transformations. Issues also lies in the robustness of CAD boolean operators which use quite complex modeling techniques. In a future work, instead of boolean operators, specific operators can be developed for an efficient implementation.

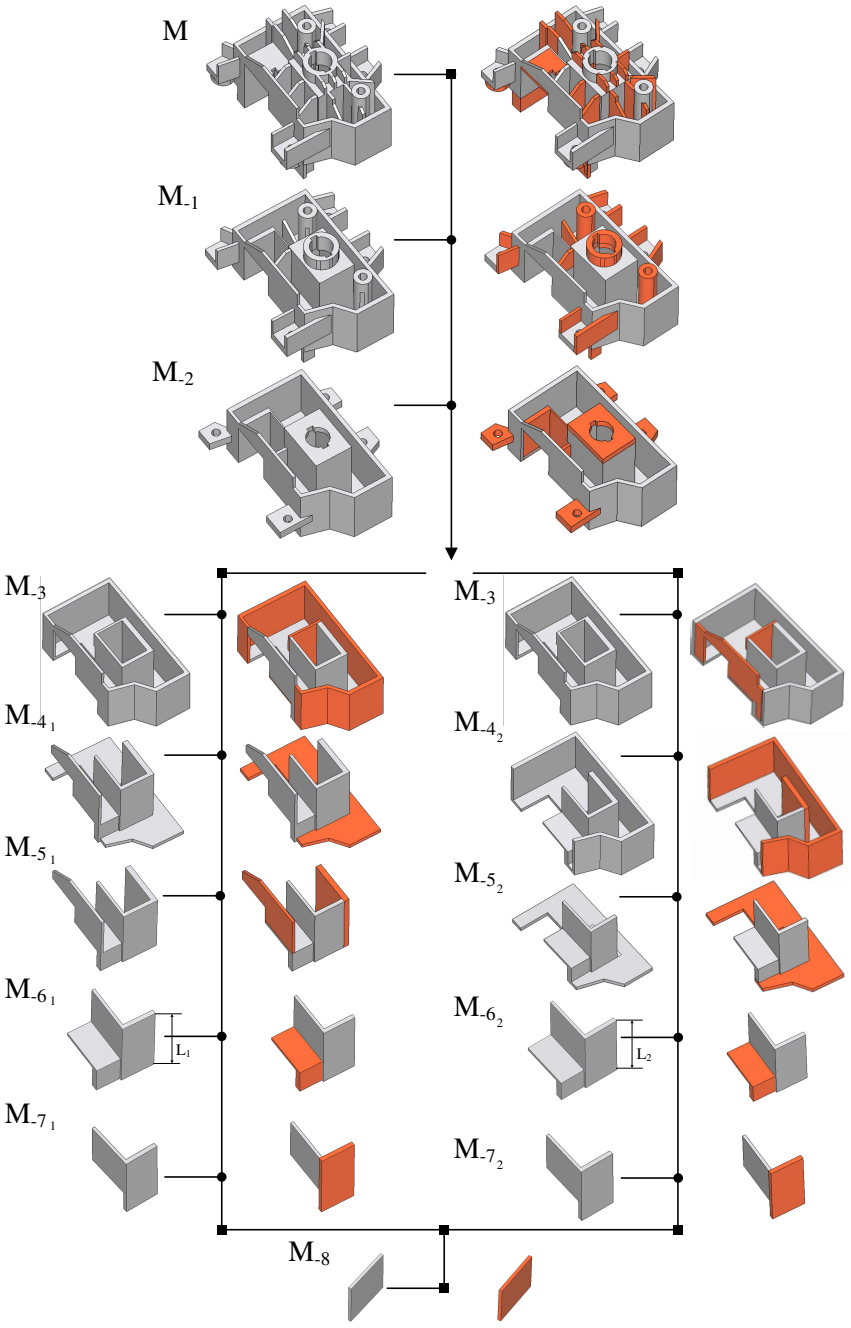


Figure 4.20: Construction graph G_D of a component. Orange sub-domains indicate the removed primitives P_i at each node of G_D. Labels M_{-j_k} indicate the step number j when 'going back over time' and the existence of variants k, if any. Arrows described the successive steps of $\mathcal{D}$. Arcs of G_D are obtained by reversing these arrows to produce construction trees. Steps M_{-6_1} and M_{-6_2} differ because of distinct lengths L_1 and L_2.

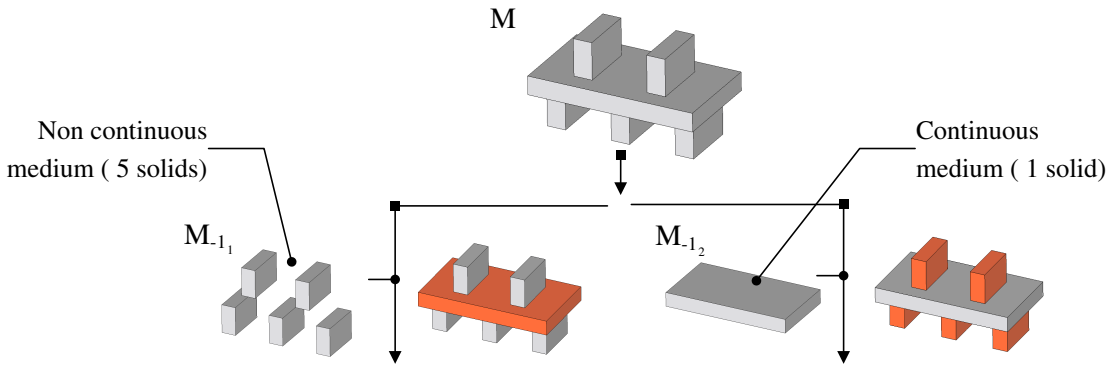


Figure 4.21: Illustration of the continuity constraint with two construction processes of an object M. The generated object M_{-1_1} is composed of 5 independent solids which represent a non continuum domain. Object M_{-1_2} contains one solid which represents a continuum domain.

Equivalence between a graph representation and a set of construction trees

Also, G_D is a compact representation of a set of construction processes. Figure 4.22 illustrates the equivalence between a set of construction trees and the graph representation G_D. There, the set of primitives removed from M_{-j} to obtain $M_{-(j+1)}$ is characterized by the edge $\alpha_{-j,-(j+1)}$ of G_D that connects these two models. The cardinality of this set of primitives is n_j. If these n_j primitives are related to different sketching planes, they must be attached to n_j different steps of a construction tree in a CAD software. Without any complementary criterion, the ordering of these n_j primitives can be achieved in $n_j!$ different ways involving n_j additional nodes and edges in G_D to represent the corresponding construction tree. Here, it is compacted into a single graph edge $\alpha_{-j,-(j+1)}$ in G_D between M_{-j} and $M_{-(j+1)}$. Furthermore, the description of this set of tree structures requires the expansion of $\alpha_{-j,-(j+1)}$ into the $n_j!$ construction tree structures ending up to M_{-j}. This modification of G_D generates new cycles in G_D between M_{-j} and $M_{-(j+1)}$. Indeed, the graph structure of G_D is a much more compact structure than the construction tree of CAD software. All the previous results and observations show that G_D is a promising basis for getting a better insight about a shape structure and evaluating its adequacy for idealizations.

4.7 Extension of the component segmentation to assembly structure segmentation

This section explains how the generative processes used for single B-Rep component can be adapted to assembly structures of several B-Rep components.

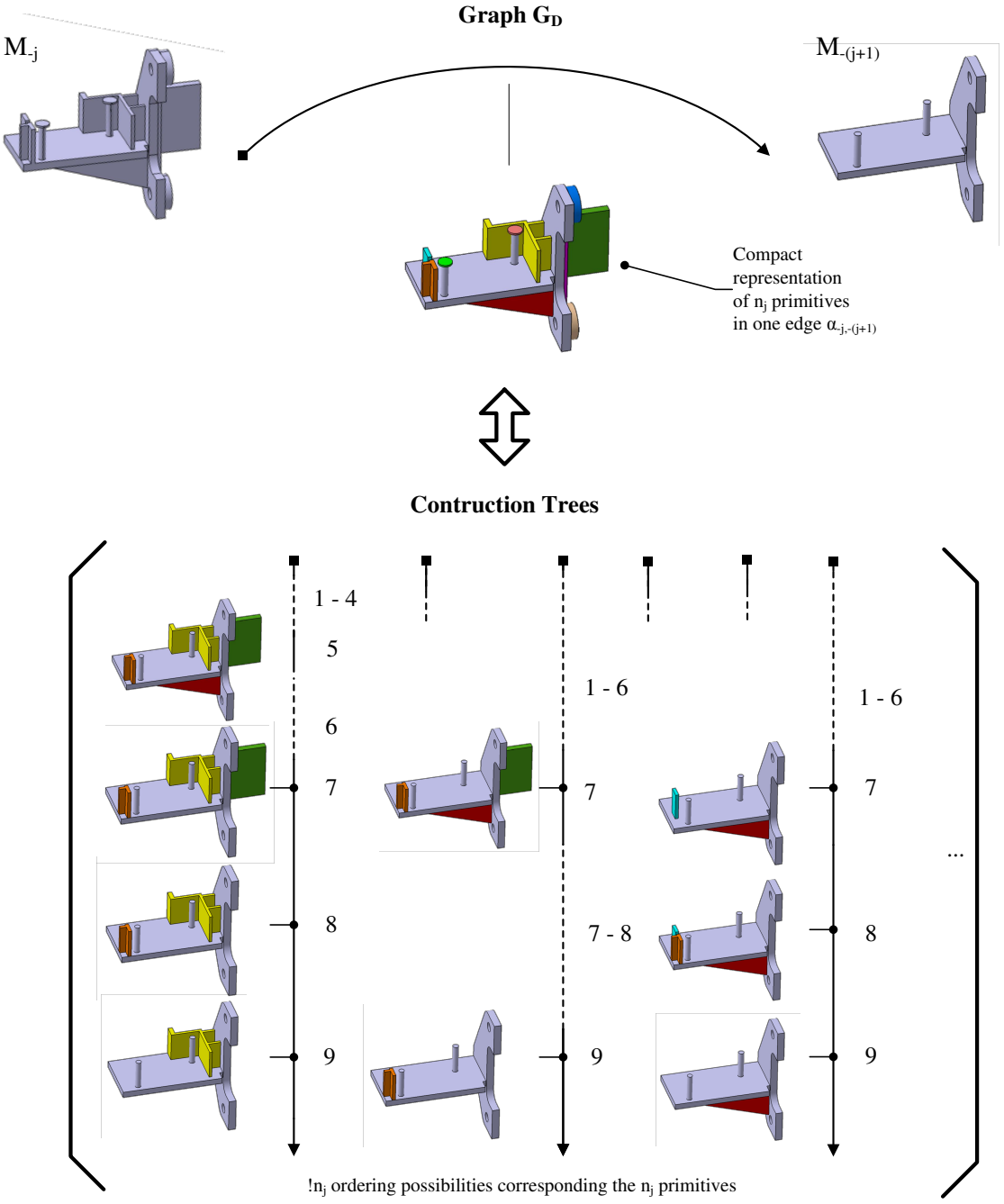


Figure 4.22: A set of CAD construction trees forming a graph derived from two consecutive construction graph nodes.

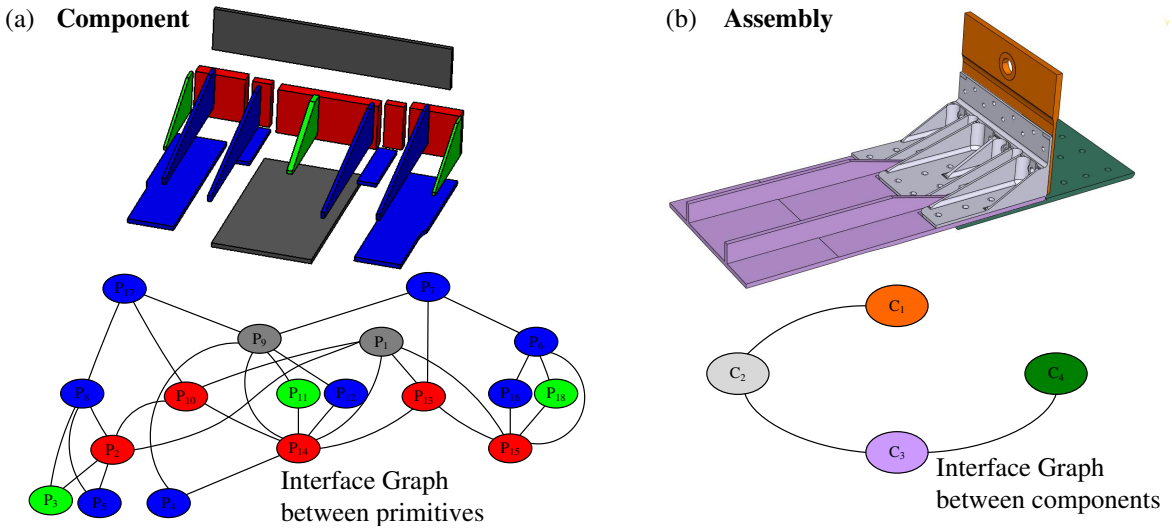


Figure 4.23: Illustration of the compatibility between the component segmentation (a) and assembly structure segmentation (b).

Equivalence between generative processes of components and assembly structures

As explained in 1.3.1, a CAD assembly structure contains a set of volume B-Rep components located in a global reference frame. The method of Jourdes et al. [JBH*14] enriches the assembly with geometric interfaces between components. Shahwan et al. [SLF*12, SLF*13] further enrich the results of Jourdes et al. with functional designation of components. As a result, the final assembly model is composed of a set of 3D solid components connected to each other through functional interfaces.

An equivalence can be made between this enriched assembly structure (see Figure 4.23b) and generative processes of components. Indeed, a solid decomposition of each component can be derived from its generative processes expressed with G_D. It provides intrinsic structures of components made of 3D solid primitives linked by geometric interfaces (see Figure 4.23a). These structures are compatible with the assembly structure also described using 3D solids and interfaces. The decomposition of each component, G_D, can be integrated into an assembly graph structure. Figure 4.24a illustrates an assembly of two components C_1 and C_2 connected by one assembly interface $I_{1,2}$. In 4.24b, each component is subdivided into two primitives $(P_{1,1}, P_{1,2})$ and $(P_{2,1}, P_{2,2})$, respectively, linked by geometric interfaces $I_{1_1,1_2}$ and $I_{2_1,2_2}$, respectively. Now, the assembly structure can be represented by a graph G_A where nodes represent the assembly components and edges contains functional assembly interfaces. Each solid decomposition of a component C_i also constitutes a graph structure G_{D_i} which can be nested into the nodes of G_A, see 4.24c, in a first place.

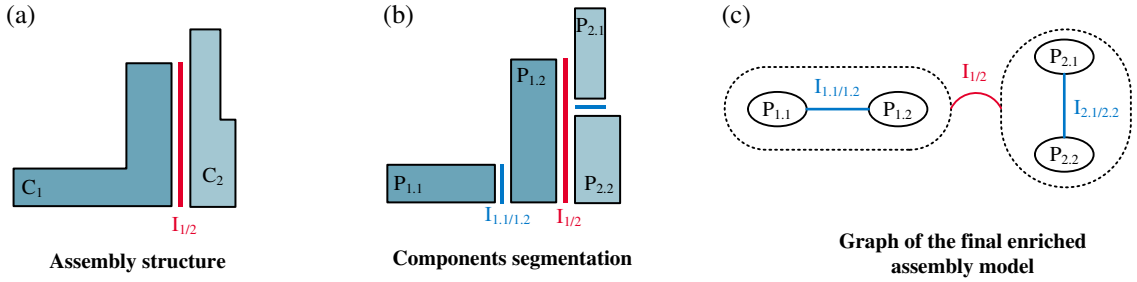


Figure 4.24: Insertion of the interface graphs between primitives obtained from component segmentations into the graph of assembly interfaces between components G_A: (a) The assembly structure with its components and assembly interfaces between these components, (b) the components segmented into primitives and interfaces between these primitives forming G_{D_i}, (c) the graph of the final enriched assembly model.

Advantages for the shape analysis of components and assemblies

The compatibility of the component segmentation with the assembly graph structure is a great benefit for the analysis algorithms dedicated to the determination of the simulation modeling hypotheses. Considering sub-domains and interfaces as input data to a general framework enabling the description of standalone components as well as assemblies, the analysis algorithms can be applied at the level of a standalone component as well as at the assembly level. This property extends the capabilities of the proposed FEA pre-processing methods and tools from the level of standalone components to an assembly level and contributes to the generation of FE analyses of assembly structures.

However, it has to be pointed out that the nesting mechanism of G_A and G_{D_i} has been briefly sketched and a detailed study is required to process configurations in which component interfaces are not exactly nested into component primitive faces, e.g., interface $I_{1,2}$ that covers faces of $P_{2,1}$ and $P_{2,2}$. Additionally, interfaces used for illustration are all of type surface, whereas interfaces between primitives can be of types surface or volume and geometric interfaces between components can be of type contact or interference. If, in both cases, this leads to common surfaces or volumes, the detailed study of these configurations is left to future work to obtain a thorough validation.

4.8 Conclusion

This chapter has described a new approach to decompose a B-Rep shape into volume sub-domains corresponding to primitive shapes, in order to obtain a description that is intrinsic to this B-Rep shape while standing for a set of modeling actions that will be used to identify idealizable sub-domains.

Construction trees and shape generation processes are common approaches to model mechanical components. Here, it has been shown that construction trees can be extracted from the B-Rep model of a component. Starting with a B-Rep object free of blends, the proposed approach processes it by iteratively identifying and removing a set of volume extrusion primitives from the current shape. The objective of this phase is to 'go back in time' until reaching root primitives of generative processes. As a result, a set of non-trivial generative processes (construction trees) is produced using the primitives obtained at the end of this first phase.

It has been shown that construction trees are structured through a graph to represent the non trivial collection of generative processes that produce the input B-Rep model. This graph contains non trivial construction trees in the sense that neither variants of extrusion directions producing the same primitive are encoded nor are the combinatorial variants describing the binary construction trees of CAD software, i.e., material addition operations that can be conducted in parallel are grouped into a single graph node to avoid the description of combinatorial combinations when primitives are added sequentially as in CAD software. Thus, each node in the construction graph can be associated with simple algorithms to generate the trivial construction variants of the input object.

The proposed method includes criteria which generate primitives with simple shapes and which ensure that the shape of intermediate objects is simplified after each primitive suppression. These properties guarantee the algorithm to be convergent.

Finally, a graph of generative processes of a B-Rep component is a promising basis to gain a better insight of a shape structure and to evaluate its adequacy for idealizations. It has been illustrated that this process can also be extended to the assembly context with the nesting of the primitive-interface structure with respect to the component-interface one. The next Chapter 5 describes how a construction graph can be efficiently used in an idealization process.

Chapter 5

Performing idealizations from construction graphs

Benefiting from the enrichment of a component shape with its construction graph, this chapter details the proposed morphological approach and the idealization process to generate idealized representations of a component shape' primitives. Based on this automated decomposition, each primitive is analyzed in order to define whether it can be idealized or not. Subsequently, geometric interfaces between primitives are taken into account to determine more precisely the idealizable sub-domains. These interfaces are further used to process the connections between the idealized sub-domains generated from these primitives to finally produce the idealized model of the initial object. Also it is described how the idealization process can be extended to assembly models.

5.1 Introduction

According to Section 1.5.4, shape transformations taking place during an assembly simulation preparation process interact with simulation objectives, hypotheses, and shape transformations applied to standalone components and to their assembly interfaces. Section 2.3 underlines the value of a shape analysis prior to the application of these transformations to characterize their relationship with geometry adaption and FEA modeling hypotheses, especially in the scope of idealization processes which need to identify the candidate regions for idealization.

As explained in Chapter 2, prior research work has concentrated on identifying idealizable areas rather than producing simple connections between sub-domains [LNP07b, SSM*10, MAR12, RAF11]. Recent approaches [CSKL04, Woo14] subdivide the input CAD model into simpler sub-domains. However, these segmentation algorithms do not aim at verifying the mechanical hypotheses of idealization processes and the identified features found do not necessarily produce appropriate solids for dimensional reduction operators. The idealization process proposed in this chapter benefits from the shape

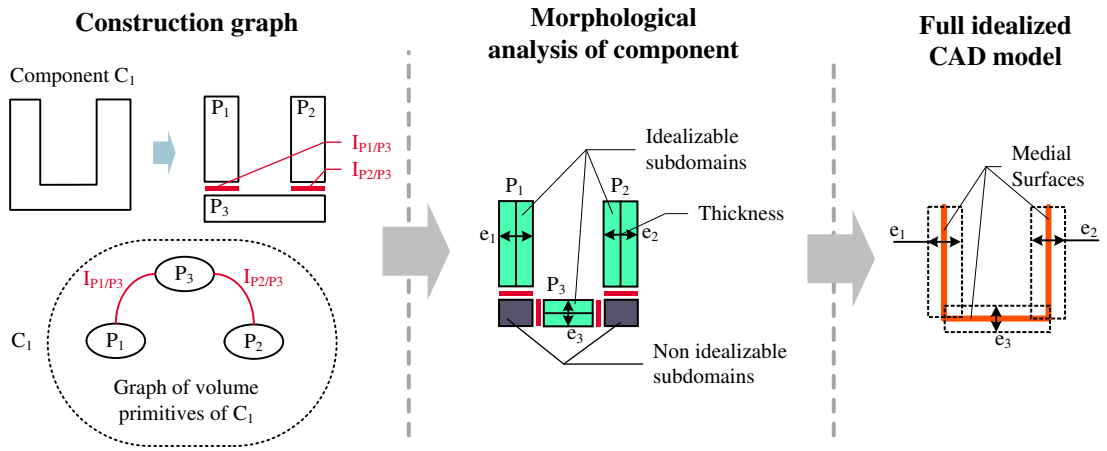


Figure 5.1: From a construction graph of a B-Rep shape to a full idealized model for FEA.

structure produced by construction graphs generated from a component as a result of the process described in Chapter 4 and containing volumes extrusion primitives which can be already suited to idealization processes. This construction graph directly offers the engineer different segmentation alternatives to test various idealization configurations.

Starting from a construction graph of a B-Rep model, this chapter describes a morphological analysis approach to formalize the modeling hypotheses for the idealization of a CAD component. This formalization leads to the generation of a new geometric structure of a component that is now dedicated to the idealization process. Figure 5.1 illustrates the overall process which is then described in the following sections. Section 5.2 explains the advantages of applying a morphological analysis on a component shape and states the main categories of morphology that needs to be located. Section 5.3 describes the general algorithm proposed to analyze the morphology of a B-Rep shape from its generative processes containing extrusion primitives. The algorithm evaluates each primitive morphology and process interfaces between these primitives to extend the morphological analysis to the whole object. Section 5.4 studies the influence of external boundary conditions and of assembly interfaces on the new component structure. There, the objective is to determine the adequacy of the proposed approach with an assembly structure. Section 5.5 illustrates the process to derive idealized models from a set of extrusion primitives and geometric interfaces.

5.2 The morphological analysis: a filtering approach to idealization processes

A common industrial principle of the FEA of mechanical structures is to process the analysis step-by-step using a top-down approach. To simulate large assembly struc-

tures, i.e., aeronautical assemblies, a first complete idealized mesh model is generated to evaluate the global behavior of the structure. Then, new local models are set up to refine the global analysis in critical areas. At each step of this methodology, the main purpose of the pre-processing task is to generate, as quickly as possible, a well suited FE model. In the industry, guidelines have been formalized to help engineers defining these FE models and correctly applying the required shape transformations. Although these guidelines are available for various simulation objectives, there are still difficulties about:

- The model accuracy needed to capture the mechanical phenomenon to be evaluated. An over-defined model would require too many resources and thus, would delay the results. In case of large assemblies, the rules set in the guidelines are very generic to make sure a global model can be simulated in practice. This way, the FEM is not really optimized. As an example, the mesh size is set to a constant value across all geometric regions of the components and is not refined in strategic areas. Section 1.4 points out current engineering practices to generate large FE assembly models;

- The application of modeling rules. The engineer in charge of the FEM creation has difficulties when identifying the regions potentially influencing or not the mechanical behavior of the structure, prior to the FEA computation results. In addition, evaluating the geometric extent of the regions to be idealized as well as determining the cross influences of geometric areas on each other are difficult tasks for the engineer because they can be performed only mentally, which is even harder for 3D objects. Section 2.3 highlights the lack of idealization operators to delimit their conditions of application.

In case of large assembly data preparation, automated tools are required (due to model size and time consuming tasks). These tools have to produce simulation models in line with the two previous challenges (model accuracy and rule-based modeling). In the following sections, we introduce approaches to deal with such challenges using morphological analysis tools. These tools support the engineers when comparing the manufactured shape of a component with the simplification and idealization hypotheses needed to meet some simulation hypotheses.

5.2.1 Morphological analysis objectives for idealization processes based on a construction graph

As introduced in Chapter 3, a major objective of this thesis aims at improving the robustness of FE pre-processing using a shape analysis of each component before the application of shape transformation operators. For a simulation engineer, the purpose is to understand the shape, support the localization of transformations, and to build

the different areas to transform in the initial CAD models. This scheme contributes to an a priori approach that illustrates the application of modeling hypotheses, especially the idealization hypotheses. In addition, this approach is purely morphological, i.e., it does not depend on discretization parameters like FE sizes.

Morphological categories identified in solid models

The first requirement of the idealization process is to identify which geometric regions of the object contain proportions representing a predefined mechanical behavior. In the scope of structural simulations, the predefined categories representing a specific mechanical behavior correspond to the major finite element families listed in Table 1.1. These categories can be listed as follows:

- Beam: a geometric sub-domain of an object with two dimensions being significantly smaller than the third one. These two dimensions define the beam cross section;
- Plate and shell: a geometric sub-domain of an object with one dimension being significantly smaller than the two others. This dimension defines the thickness of the plate or the shell, respectively;
- 3D thick domain: a geometric sub-domain that does not benefit from any of the previous morphological properties and that should be modeled with a general 3D general continuum medium behavior.

From a geometric perspective, the principle of the idealization process corresponds to a dimensional reduction of the manifold representing a sub-domain of the solid object, as defined in 1.2.1. A detailed description of the idealization hypotheses is available in Section 1.4.2. To construct fully idealized models derived from an object M, geometric connection operations between idealized sub-domains must be performed. As stated in Chapter 2, automated geometric transformations do not produce accurate results in connections areas. The main issue lies in the determination of the geometric boundaries where the connection operators must process the idealized sub-domains (see Figure 2.8). Currently, the engineer applies these operators manually to define the direction and the extent of the connection areas (see Figure 1.15 illustrating a manual process to connect medial surfaces). The proposed idealization process benefits from an enriched initial component model with a volume segmentation into sub-domains and into interfaces between them that contain new geometric information to make the connections operators robust. This segmentation gives the engineer, a visual understanding of the impact of simulation hypotheses on the component geometry. Additionally, the proposed analysis framework identifies the regions that can be regarded as details, independently of the resolution method. These regions represents areas having no significant mechanical influence with respect to the morphological category of their neighboring regions.

In case of assembly structures, the categories presented above are still valid at the difference that the identified geometric domains are not anymore a sub-domain restricted to a single solid. Indeed, a group of components can be considered as a unique beam, plate, or shell element. In this case, if the interfaces between a group of connected sub-domains from different components have been defined as mechanically rigid connections by the engineer, this group of sub-domains can also be identified as a unique continuum, hence distinguishing components is not necessary anymore. The effects of assembly interfaces are detailed in Section 5.4.

Adequacy of generative construction graphs with respect to the morphological analysis

A shape decomposition is a frequent approach to analyze and structure objects for FEA requirements. Section 2.4 has highlighted the limits of current morphological analysis methods and underlined the need of more robust automatic techniques adapted to FE model generation. The proposed approach uses the generative construction graph G_D to perform a morphological analysis of CAD components adapted to FEA requirements. Section 4.2 has addressed the advantages of construction graphs to structure the shape of a component. It proposes a compact shape decomposition of primitives containing a fair amount of information that is intrinsic to this shape. The geometric structure of G_D with sub-domains and associated interfaces is close to the structure described in Section 5.2.1 with regions candidate to idealization. G_D also offers various construction processes which enable the engineer to construct and study various simulation models that can be derived from the same component using different construction processes.

Generating the idealization of an object M from a set of primitives, obtained from its construction graph G_D, is more robust compared to the prior idealization methods for three reasons:

- The information contained in G_D is intrinsic to the definition of primitives. Each maximal primitive $P_i \in G_D$ and its associated interfaces determine both the 3D location of the idealized representation of P_i and its connections with its neighboring idealized primitives;

- The effective use of connections between sub-domains to be idealized. A taxonomy of categories of connections can be defined. This classification determines the most suitable geometric operator to process each connection. Currently, the idealization process still relies on the engineer's expertise to manage complex connections whereas a CAD or a CAE software is bound to much simpler connections;

- Shape modification processes of components. When a component shape modification is performed, only the impacted primitives have to be revised in its

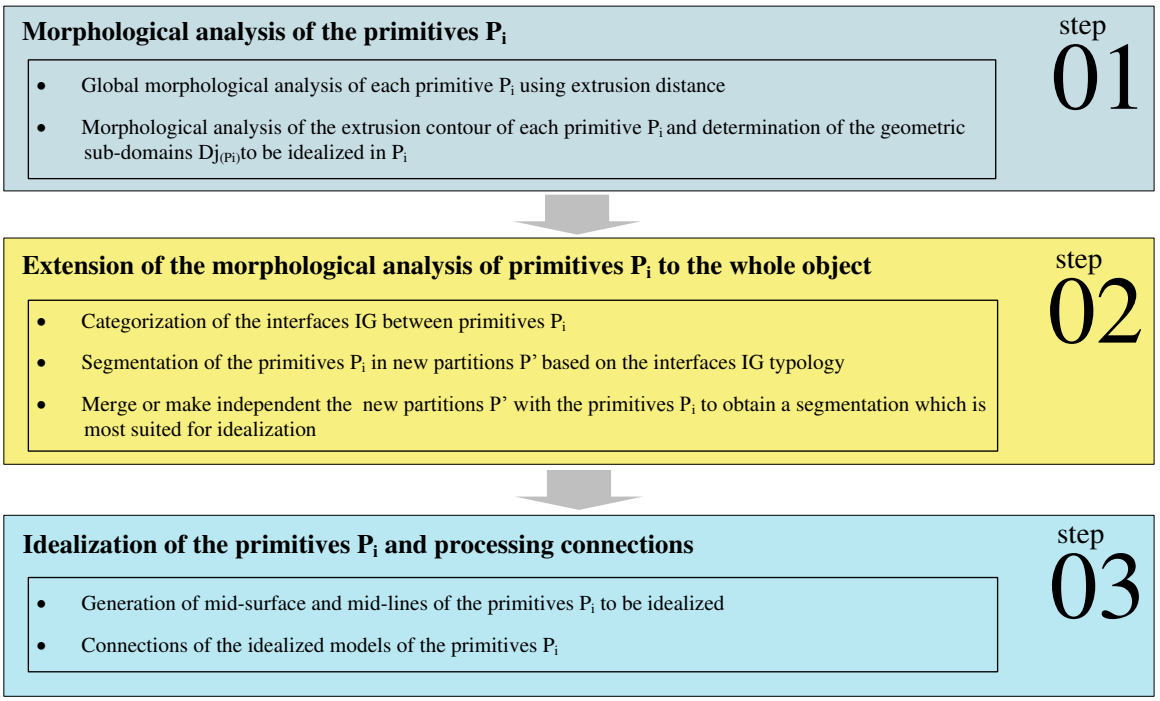


Figure 5.2: Global description of an idealization process.

construction graph. Therefore, the idealization process can be locally updated and does not have to be restarted over from its shape decomposition.

The next section details the structure of the proposed idealization process from the exploitation of the geometric information provided with the construction graph G_D of a component to the generation of its idealized models.

5.2.2 Structure of the idealization process

The idealization process of an object M is based on morphological analysis operations, idealization transformations and connections between idealized sub-domains. Its objective is to produce the idealized model, denoted by M_I, of the initial object M.

Figure 5.2 contains a comprehensive chart illustrating the various steps of the proposed idealization process. In a first step, the decomposition of M into primitives P_i, described in the graph G_D, leads to a first morphological analysis of each primitive P_i. For each extrusion primitive, this morphological analysis determines whether P_i has a morphology of type plate, shell, or a morphology of type thick 3D solid (see Section 5.2 describing the morphology categories). This first step is described in Section 5.3.1. Then, a second morphological analysis is applied to each primitive P_i to determine whether or not it can be subdivided into sub-domains D_{ij}, morphologically different from the one assigned to P_i. This analysis is performed using the 2D extrusion contour

of the primitive P_i. The resulting decomposition of P_i generates new interfaces I_G, integrated in G_D.

In a second phase, using a typology of connections between the different categories of idealizations, the interfaces I_G between D_{ij} are used to propagate and/or update the boundary of the primitives P_i. This step, described in Section 5.3.3, results in the decomposition of the object M into sub-domains with a morphology of type beam, plate/shell, or 3D thick solid. The third phase consists in generating the idealization of each primitive P_i and then, it connects the idealized domains of P_i using the typology and the location of the interfaces I_G. During this operation, the morphology of each P_i, combined with the typology of each of its interfaces I_G, is used to identify regions to be considered as details, independently from any FE size. The generation of an idealized model is described in Section 5.5.

Overall, the different phases illustrated in Figure 5.2 fit into an automated process. The engineer can be involved in it to select some connection model between idealized sub-domains or some boundary adjustment category of idealized sub-domain depending on its types of connections. The next sections detail each step of the proposed idealization process.

5.3 Applying idealization hypotheses from a construction graph

The purpose of this section is to illustrate how the construction graph G_D of an object M obtained with the algorithm described at Section 4.5.2 can be used in shape idealization processes. In fact, idealization processes are high level operations that interact with the concept of detail because the idealization of sub-domains, i.e., P_i obtained from G_D, triggers their dimensional reduction, which, in turn, influences the shape of areas around I_Gs, the geometric interfaces between these sub-domains. Here, the proposed approach is purely morphological, i.e., it does not depend on discretization parameters like FE sizes. It is divided into two steps. Firstly, each P_i of G_D is evaluated with respect to an idealization criterion. Secondly, according to I_Gs between P_is, the 'idealizability' of each P_i is propagated in G_D through the construction graph up to the shape of M. As a result, an engineer can evaluate effective idealizable areas. Also, it will be shown how variants of construction trees in G_D can influence an idealization process. Because the idealization process of an object is strongly depending on the engineer's know-how, it is the principle of the proposed approach to give the engineer access to the whole range of idealization variants. Finally, some shape details will appear subsequently to the idealization process when the engineer will define FE sizes to mesh the idealized representation of M.

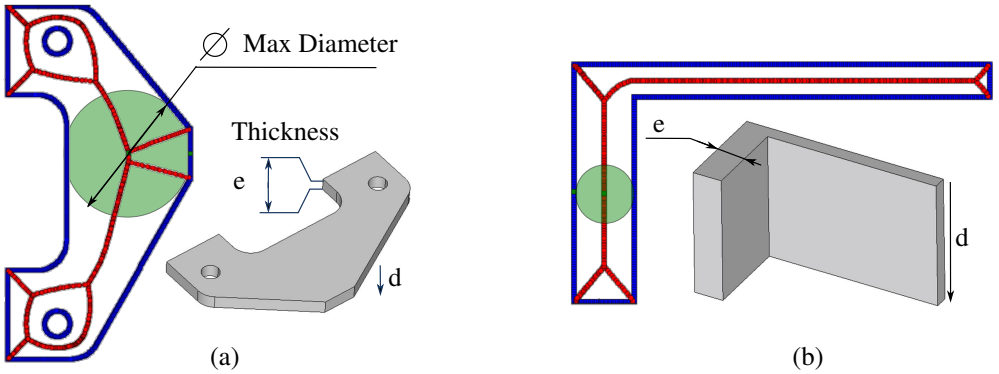


Figure 5.3: Determination of the idealization direction of extrusion primitives using a 2D MAT applied to their contour. (a) Configuration with an extrusion distance (i.e., thickness d = e) much smaller than the maximal diameter obtained with the 2D MAT on the extrusion contour, the idealization direction corresponds to the extrusion direction (b) Configuration with an extrusion distance much larger than the maximal diameter obtained with the 2D MAT on the extrusion contour, the idealization direction is included in the extrusion contour.

5.3.1 Evaluation of the morphology of primitives to support idealization

Global morphological analysis of each primitive P_i

In a first step, each primitive P_i extracted from G_D is subjected to a morphological analysis to evaluate its adequacy for idealization transformation into a plate or a shell. Because the primitives are all extrusions and add material, analyzing their morphology can be performed with a MAT [MAR12, RAM*06, SSM*10].

A MAT is particularly suited to extrusion primitives having constant thickness since it can be applied in 2D. Furthermore, it can be used to decide whether or not sub-domains of P_i can be assigned a plate or shell mechanical behavior. In the present case, the extrusion primitives obtained lead to two distinct configurations (see Figure 5.3). Figure 5.3a shows a configuration with a thin extrusion, i.e., the maximal diameter Φ obtained with the MAT from P_i's contour is much larger than P_i's thickness defined by the extrusion distance d. Then, the idealized representation of P_i would be a surface parallel to the base face having P_i's contour. Figure 5.3b shows a configuration where the morphology of P_i leads to an idealization that would be based on the content of the MAT because d is much larger than Φ.

To idealize a sub-domain in mechanics [TWKW59], a commonly accepted reference proportion used to decide whether a sub-domain is idealizable or not is a ratio of ten between the in-plane dimensions of the sub-domain and its thickness, i.e., $x_r = 10$. Here, this can be formalized with the morphological analysis of P_i obtained from the MAT using: $x = \max((\max\Phi/d), (d/\max\Phi))$. Consequently, the ratio x is applicable for all morphologies of extrusion primitives.

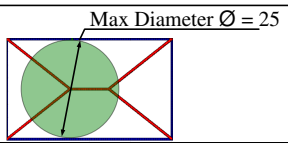
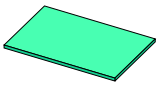
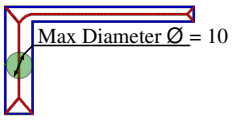
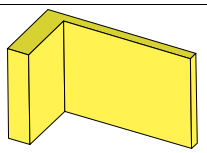
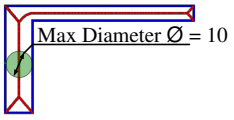
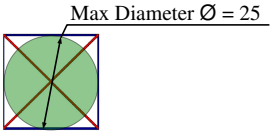
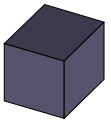
Primitive d: extrusion distance	**MAT 2D**	$d < \max Ø$ $\llcorner x = \frac{\max Ø}{d}$ $d > \max Ø$ $\llcorner x = \frac{d}{\max Ø}$ **Morphology** 0 — 3 — 10 x_u — x_r Ratio x
(a) ↓d = 1.5		$1.5 < <25$ $\llcorner x = \frac{25}{1.5} = 16.6$ 
(b) d = 35		$35 > 10$ $\llcorner x = \frac{35}{10} = 3.5$ 
(c) ↓d = 2		$2 < 10$ $\llcorner x = \frac{10}{2} = 5$ 
(d) d = 25		$25 = 25$ $\llcorner x = \frac{25}{25} = 1$ 

Table 5.1: Categorization of the morphology of a primitive using a 2D MAT applied to the contour of extrusion primitives. Violet indicates sub-domains that cannot be idealized as plates or shells (see component d), green ones can be idealized (see component a) and yellow ones can be subjected to user decision (see component b and c).

Because idealization processes are heavily know-how dependent, using this reference ratio as unique threshold does not seem sufficient to help an engineer analyze sub-domains, at least because x_r does take precisely into account the morphology of P_i's contour. To let the engineer tune the morphological analysis and decide when P_i can/cannot be idealized a second, user-defined threshold, $x_u < x_r$, is introduced that lies in the interval $]0, x_r[$. Figure 5.3b illustrates a configuration where the morphological analysis does not produce a ratio $x > x_r$ though a user might idealize P_i as a plate.

Let $x_u = 3$ be this user-defined value, Table 5.1 shows the application of the 2D MAT to the contours of four extrusion primitives. This table indicates the three categories made available to the engineer to visualize the morphology of P_i. Primitives with a ratio of $x > x_r$, e.g., primitive (a), are considered to be idealizable and are colored in green. Primitives with a ratio of $x_u < x < x_r$, e.g., primitives (b) and (c), are subjected to a user's decision for idealization and are colored in yellow. Finally, primitives with a ratio $x < x_u$, e.g., primitive(d), indicate sub-domains that cannot be idealized and are colored in violet.

Figure 5.4 illustrates the evaluation of the morphology of the primitives of a component prior to its idealization. The component has been initially segmented into 15 extrusion primitives using the algorithm presented at Section 4.5.2. Then, the 2D MAT has been applied to the extrusion contour of each primitive to determine the maximal diameter Φ. Finally, this diameter is compared to the extrusion distance of the corresponding primitive to determine the ratio x. Three morphological evaluations are presented in Figure 5.4 that correspond to different values of the thresholds x_u and x_r, which are set to (a) $(3, 10)$, (b) $(6, 10)$, and (c) $(2, 10)$, respectively.

Figure 5.5 shows the result of the interactive analysis the user can perform from the graphs G_D obtained with the components analyzed in Figures 5.5a, b, c, and d. It has to be mentioned that the morphological analysis is applied to G_D rather than to a single construction tree structure so that the engineer can evaluate the influence of $\mathcal{D}$ with respect to the idealization processes. However, the result obtained on the component in Figure 4.20 shows that the variants in G_D have no influence with respect to the morphological analysis criterion, in the present case. Consequently, Figure 5.5 displays the morphological analysis obtained from the variant M_{-j_2} in Figure 4.20. Results on components in 5.5a, c also show the clear use of this criterion because some non-idealizable sub-domains (see indications in Figure 5.5 regarding violet sub-domains) are indeed well proportioned to be idealized with beams.

Now, considering the morphological classification of sub-domains stated in Section 5.2.1, this first morphological analysis of P_i acts as a necessary condition for P_i to fall into a category of:

- *plate/shell* but P_i can contain sub-domains D_{ij} where Φ can get small enough to produce a *beam*-like shape embedded in P_i, see Figure 5.6a. In any case, P_i

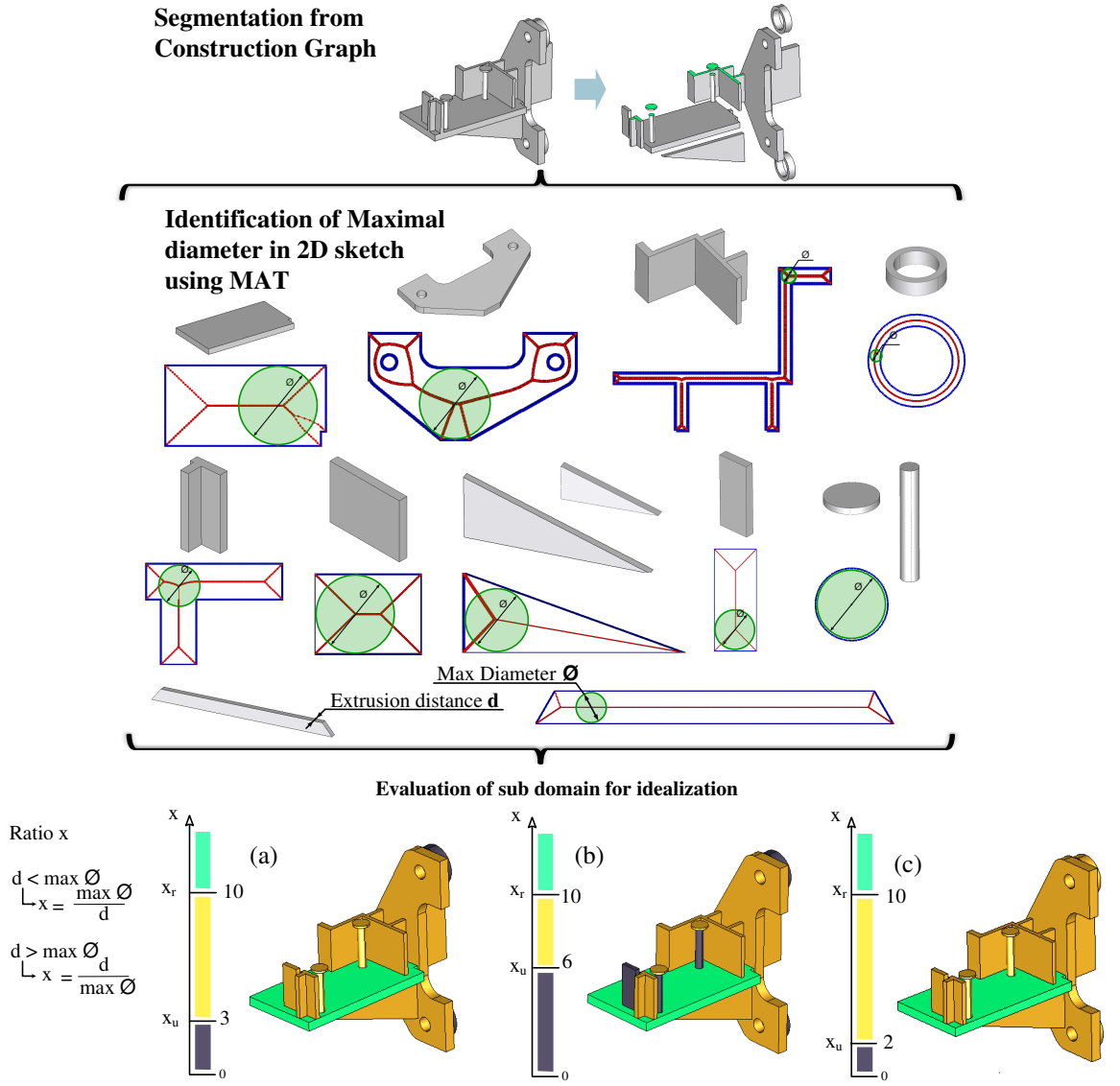


Figure 5.4: Example of the morphological evaluation of the extrusion primitives extracted during the construction process of a component. Violet indicates sub-domains that cannot be idealized as plates or shells, green ones can be idealized and yellow ones are up to the user's decision.

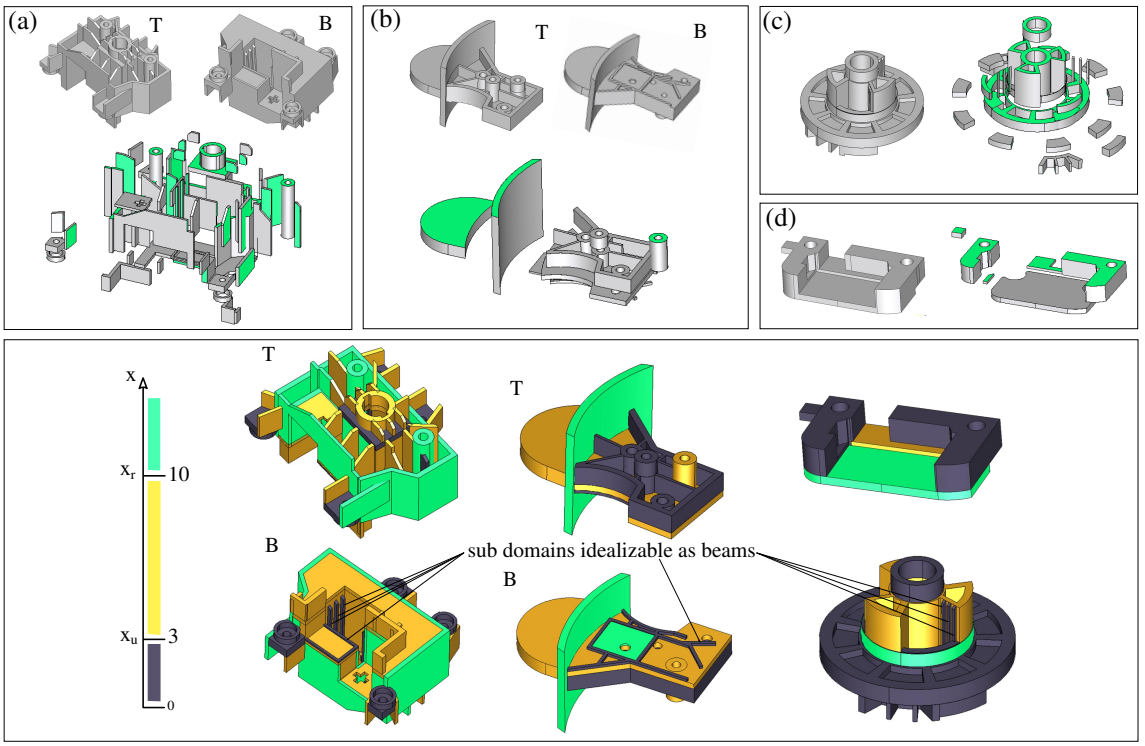


Figure 5.5: Idealization analysis of components. T and B indicate Top and Bottom views of the component, respectively. The decomposition of a is shown in Figure 4.20 and decompositions of b, c and d are shown in Figure 4.19. Violet indicates sub-domains that cannot be idealized as plates or shells, green ones can be idealized and yellow ones can be subjected to user's decisions.

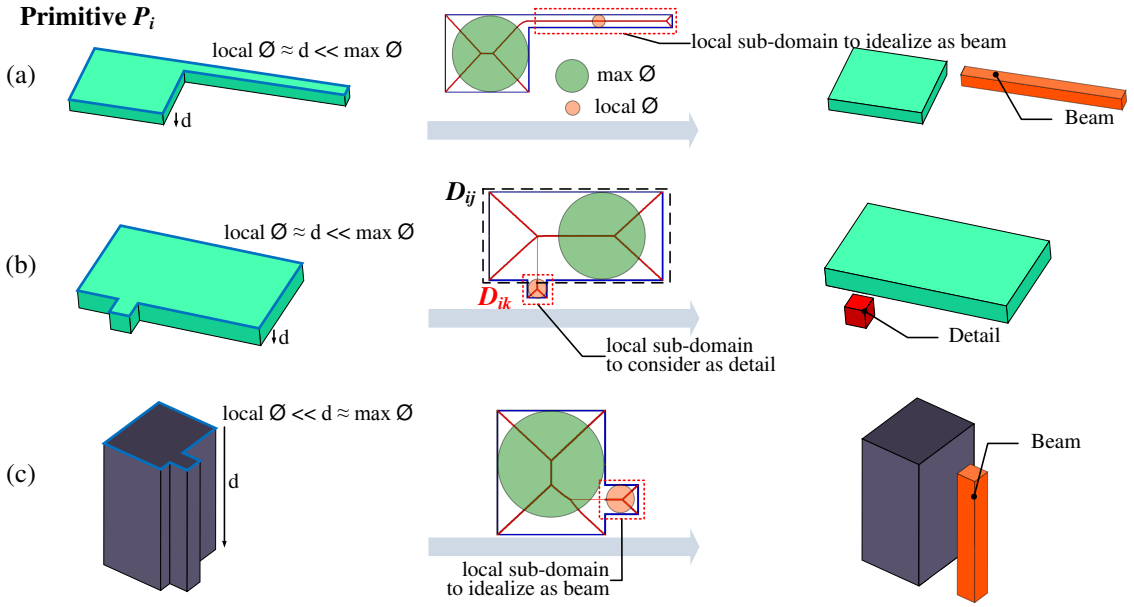


Figure 5.6: Illustration of primitives' configurations containing embedded sub-domains D_{ik} which can be idealized as beams or considered as details. (a) Primitive morphologically identified as a *plate* which contains a beam sub-domain, (b) primitive morphologically identified as a *plate* which contains a volume sub-domain to be considered as a detail, (c) primitive morphologically identified as a *thick domain* which contains a beam sub-domain

cannot contain a sub-domain of type 3D thick domain because the dominant sub-domain D_{ij} is morphologically a *plate/shell*. If there exists a sub-domain D_{ik}, adjacent to D_{ij}, such that $x < x_u$, i.e., it is morphologically *thick*, D_{ik} is not mechanically of type *3D thick domain* because it is adjacent to D_{ij}. Indeed, D_{ik} can be regarded as a detail compared to D_{ij} since the thickness of D_{ij} will be part of the dimensional reduction process. Figure 5.6b shows an example of such configuration;

- *3D thick domain* because P_i contains at least one dominant sub-domain D_{ij} of this category. However, it does not mean that P_i does not contain other sub-domains D_{ik} that can be morphologically of type *plate/shell* or *beam*. Figure 5.6b illustrates a beam embedded in a *3D thick domain*.

Indeed, all green sub-domains and yellow ones validated by the engineer can proceed with the next step of the morphological analysis. Similarly, violet sub-domains cannot be readily classified as *non idealizable*. Such configurations show that the classification described in Section 5.2.1 has to take into account the relative position of sub-domains D_{ij} of P_i and they are clearly calling for complementary criteria that are part of the next morphological analysis where P_i needs to be decomposed into sub-domains D_{ij} to refine its morphology using information from its MAT.

Determination of geometric sub-domains D_{ij} to be idealized in P_i

Then, in a second step, another morphological analysis determines in each primitive P_i if some of its areas, i.e., sub-domains D_{ij}, can be associated with beams and, therefore, admit further dimensional reduction. Indeed, the previous ratio x determines only one morphological characteristic of a sub-domain D_{ij}, i.e., the dominant one, of P_i because the location of the MAT, where x is defined, is not necessarily reduced to a point. For example, Figure 5.7 illustrates a configuration where x holds along a medial edge of the MAT of the extrusion contour. Similarly to the detail removal using MAT conducted by Armstrong et al. [Arm94], a new ratio y is introduced to compare the length of the medial edge to the maximal disk diameter along this local medial edge. The parameter y is representative of a local elongation of P_i in its contour plane and distinguishes the morphology of type beam located inside a morphology of type plate or shell when the starting configuration is of type similar to Figure 5.3a. If P_i is similar to Figure 5.3b, then the dominant D_{ij} is of type *beam* if x appears punctually or of type *plate/shell* if x appears along a medial edge of the MAT of the extrusion contour of P_i.

Appendix C provides two Tables C.1 and C.2 with 18 morphological configurations associated with a MAT medial edge of a primitive P_i. The two tables differ according to whether the idealization direction of P_i corresponds to the extrusion direction, see Table C.1 (type similar to Figure 5.3a), or whether the idealization direction of P_i is included in the extrusion contour, see Table C.2 (type similar to Figure 5.3b). The reference ratio x_r and user ratio x_u are used to specify, in each table, the intervals of morphology differentiating beams, plates or shells and 3D thick domains. Therefore, nine configurations are presented in Table C.1 illustrating the elongation of the extrusion contour of P_i. Table C.1 allows both the elongation of the extrusion distance and the elongation of the extrusion contour, this produces also nine configurations. These tables illustrates 18 morphological possible configurations when the medial edge represents a straight line with a constant radius for the inscribed circles of the MAT. Other configurations can be found when the medial edge is a circle, or more generally, a curve or when the radius is changing along the medial edge. These configurations have not been studied in detail and are left for future work.

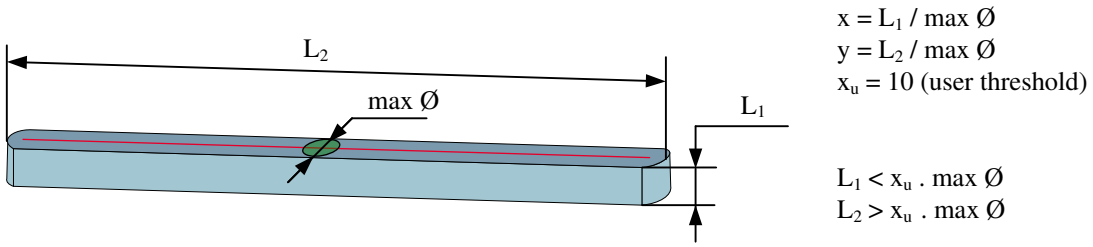


Figure 5.7: Example of a beam morphology associated with a MAT medial edge of a primitive P_i.

Tables C.1, C.2 of Appendix C represents a morphological taxonomy associated with one segment of the MAT of P_i. Because the extrusion contour of P_i consists in line segments and arcs of circles, the associated MAT has straight and curvilinear medial edges which can be categorized as follows:

1. Medial edges with one of their end point located on the extrusion contour of P_i, the other one being connected to another medial edge;

2. Medial edges with their two end points connected to other medial edges. In the special case of a segment having no end point, e.g., when the extrusion contour is a circular ring, its MAT reduces to a closed circle and falls into this category.

Segments of category 1 are deleted and the morphological analysis focuses on the segments of category 2 which are noted S_{ij}. On each of these edges, the ratio y includes a maximum located at an isolated point or it is constant along the entire edge. y_{max} represents this maximum and is assigned to the corresponding medial edge, S_{ij}. The set of edges S_{ij} is automatically classified using the taxonomy of Tables C.1, C.2 or some of them can be specified by the engineer wherever $y_u < y < y_r$. This is the interactive part left to the engineer to take into account his, resp. her, know-how.

P_i is segmented based on the changes in the morphological classification of the edges S_{ij}. This decomposition generates a set of sub-domains D_{ij} of each primitive P_i. These sub-domains D_{ij} are inserted with their respective morphological status and their geometric interfaces in the graph G_D. Figure 5.8 summarizes the different phases of the morphological analysis of each extrusion primitive P_i extracted from the construction graph G_D of on object M. Because each sub-domain D_{ij} is part of only one primitive P_i, it can also be considered as a new primitive P_k. To reduce the complexity in the following process, the sub-domains D_{ij} are regarded as independent primitives P_k.

These results are already helpful for an engineer but it is up to him, or her, to evaluate the mechanical effect of I_Gs between primitives P_i. To support the engineer in processing the stiffening effects of I_Gs, the morphological analysis is extended by a second step described as follows.

5.3.2 Processing connections between 'idealizable' sub-domains D_{ij}

The morphological analysis of standalone primitives P_i is the first application of G_D. Also, the decomposition obtained can be used to take into account the stiffening effect of interfaces I_G between P_i or, more generally, between D_{ij}, when P_i[1] are iteratively

[1]From now on P_i designates sub-domains that correspond to primitives P_i obtain from the segmentation of M or one subdivision domain D_{kj} of a primitive P_k decomposed after the morphological analysis described at Section 5.3.1.

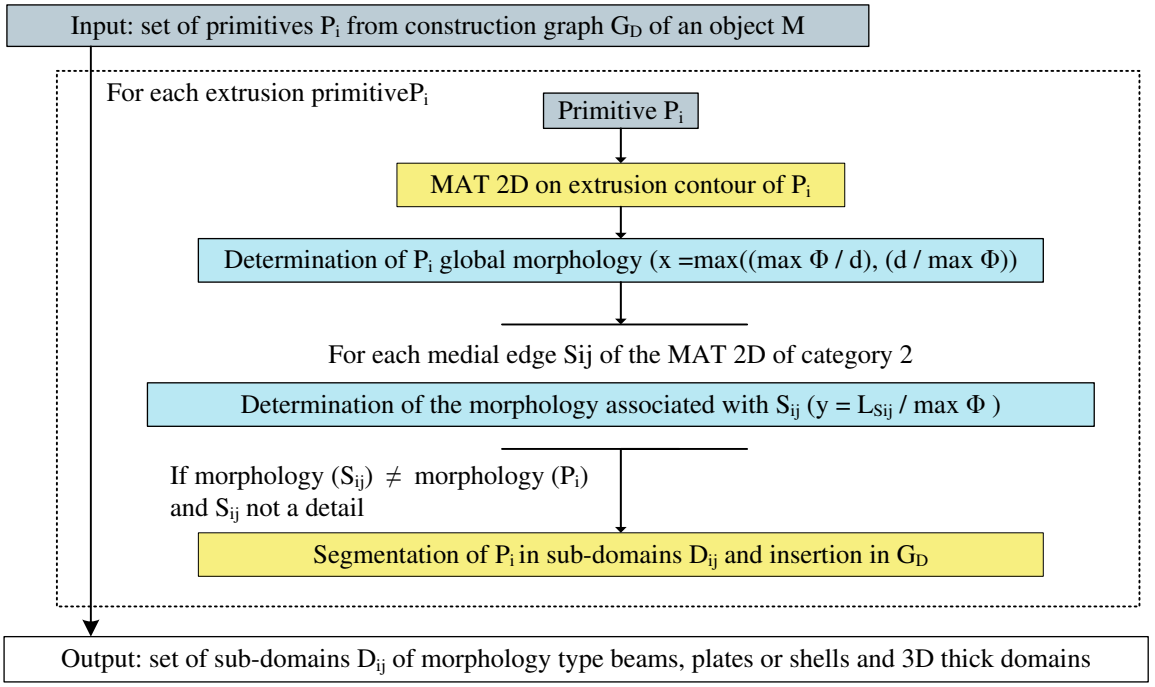


Figure 5.8: Synthesis of the process to evaluate the morphology of primitives P_i.

merged together along their I_G up to obtain the whole object M. As a result, new sub-domains will be derived from the primitives P_i and the morphological analysis will be available on M as a whole, which will be easier to understand for the engineer. To this end, a taxonomy of connections between extrusion sub-domains is mandatory.

Taxonomy of connections between extrusion sub-domains to be idealized

This taxonomy, in case of "plate sub-domain connections", is summarized in Figure 5.9a. It refers to parallel and orthogonal configurations for simplicity but these configurations can be extended to process a larger range of angles, i.e., if Figure 5.9 refers to interfaces I_G of surface type, these configurations can be extended to interfaces I_G of volume type when the sub-domains S_1 and S_2 are rotated w.r.t. each other. More specifically, it can be noticed that the configuration where I_G is orthogonal to the medial surfaces of S_1 and S_2 both is lacking of robust solutions [Rez96, SSM*10] and other connections can require deviation from medial surface location to improve the mesh quality. Figure 5.18c illustrates such configurations and further details are given in Section 5.5.2.

Figure 5.9 describes all the valid configurations of I_G between two sub-domains S_1 and S_2 when a thickness parameter can be attached to each P_i, which is presently the case with extrusion primitives.

Figure 5.9a depicts the four valid configurations: named type (1), (2), (3), (4). These configurations can be structured into two groups: type (1) and type (4) form the

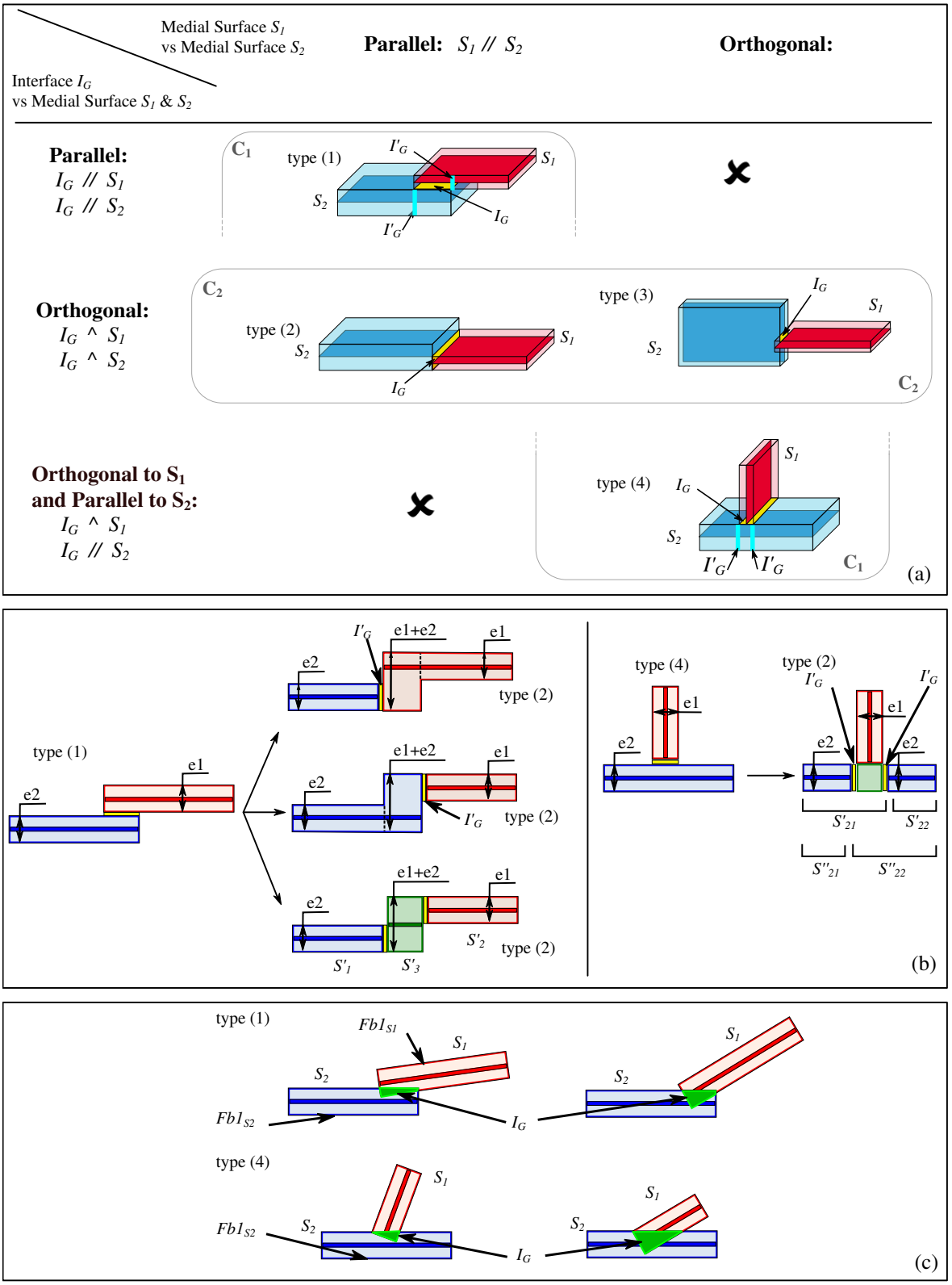


Figure 5.9: (a) Taxonomy of connections between extrusion sub-domains P_i. (b) Decomposition of configurations type(1) and type(4) into sub-domains P_i showing that the decomposition produced reduces to configurations type (2) only. (c) example configurations of types (1) and (4) where S_1 and S_2 have arbitrary angular positions that generate volume interfaces I_G where base faces $F_{b1_{S1}}$ and $F_{b1_{S2}}$ are intersection free in configuration type (1) and $F_{b1_{S2}}$ only is intersection free in configuration type (4).

group C_1, and (2) and type (3) form the group C_2. Figure 5.9b illustrates the effect of the decomposition of configurations type (1) and type (4) that produces configurations (2) only.

Reduced set of configurations using the taxonomy of connections

Configuration type (1) of C_1 is such that the thicknesses e_1 and e_2 of S_1 and S_2 respectively, are influenced by I_G, i.e., their overlapping area acts as a thickness increase that stiffens each of them. This stiffening effect can be important to be incorporated into a FE model as a thickness variation to better fit the real behavior of the corresponding structure. Their overlapping area can be assigned to either S_1 or S_2 or form an independent sub-domain with a thickness $(e_1 + e_2)$. If S_1 and S_2 are rotated w.r.t. each other and generate a volume I_G, the overlapping area still exists but behaves with a varying thickness. Whatever the solution chosen to represent mechanically this area, the sub-domains S_1 and S_2 get modified and need to be decomposed. The extent of S_2 is reduced to produce S_2' now bounded by I_G'. Similarly, the extent of S_1 is reduced to S_1' now bounded by another interface I_G'. A new sub-domain S_3' is created that contains I_G and relates to the thickness $(e_1 + e_2)$ (see Figure 5.9b). Indeed, with this new decomposition I_G is no longer of interest and the new interfaces I_G' between the sub-domains S_i' produce configurations of type (2) only.

Similarly, configuration (4) is such that S_2 can be stiffened by S_1 depending on the thickness of S_1 and/or the 2D shape of I_G (see examples in Figure 5.11). In this case, the stiffening effect on S_2 can partition S_2 into smaller sub-domains and its I_G produces a configuration of type (2) with interfaces I_G' when S_2 is cut by S_1. The corresponding decomposition is illustrated in Figure 5.9b and Figure 5.10. This time, I_G is still contributing to the decomposition of S_1 and S_2 but S_2 can be decomposed in several ways $(S'_{2_1},\ S'_{2_2})$ or $(S''_{2_1},\ S''_{2_2})$ producing interfaces I_G'. Whatever, the decomposition selected to represent mechanically this area, the key point is that I_G' located on the resulting decomposition are all of same type that corresponds to configuration (2).

Configuration (1) reduces the areas of S_1 and S_2 of constant thicknesses e_1 and e_2, which can influence their 'idealizability'. Configuration (4) reduces the area of S_2 of thickness e_2 but it is not reducing that of S_1, which influences the 'idealizability' of S_2 only. As a result, it can be observed that processing configurations in C_1 produce new configurations that always belong to C_2. Now, considering configurations in C_2, none of them is producing stiffening effects similar to C_1. Consequently, the set of configurations in Figure 5.9a is a closed set under the decomposition process producing the interfaces I_G'. More precisely, there is no additional processing needed for C_2 and processing all configurations in C_1 produces configurations in C_2, which outlines the algorithm for processing iteratively interfaces between P_i and shows that the algorithm always terminates.

Figure 5.9a and b refers to interfaces I_G of surface type. Indeed, G_D can produce

interfaces of volume type between P_i. This is equivalent to configurations where S_1 and S_2 departs from parallel or orthogonal settings as depicted in Figure 5.9. Such general configurations can fit into either set C_1 or C_2 as follows. In the 2D representations of Figure 5.9a, b, the outlines of S_1 and S_2 define the base faces F_{b1} and F_{b2} of each P_i. What distinguishes C_1 from C_2 is the fact that configurations (1) and (4) each, contains at least S_2 such that one of its base face ($F_{b1_{S2}}$ in Figure 5.9c) does not intersect S_1 and this observation applies also for S_1 in configuration (1) ($F_{b1_{S1}}$ in Figure 5.9c). When configurations differ from orthogonal and parallel ones, a first subset of configurations can be classified into one of the four configurations using the distinction observed, i.e., if a base face of either S_1 or S_2 does not intersect a base face of its connected sub-domain, this configuration belongs to C_1 and if this property holds for sub-domains S_1 and S_2 both, the corresponding configuration is of type (1). Some other configurations of type (4) exist but are not detailed here since the purpose of the above analysis is to show how the reference configurations of Figure 5.9a can be extended. The completeness of configurations has not been entirely investigated yet.

5.3.3 Extending morphological analyses of P_i to the whole object M

Now, the purpose is to use the stiffening influence of some connections as analyzed in Section 5.3.2 to process all the I_G between P_i, to be able to propagate and update the 'idealizability' of each P_i when merging P_is. This process ends up with a new subdivision of some P_i as described in the previous section and a decomposition of M into a new set of sub-domains P_i[2], each of them having an evaluation of its 'idealizability' so that the engineer can evaluate more easily the sub-domains he, or she, wants to effectively idealize.

The corresponding algorithm can be synthesized as follows (see algorithm 2). The principle of this algorithm is to classify I_G between two P_i such that if I_G belongs to C_1 (configurations 1 and 4 in algorithm 2), it must be processed to produce new interface(s) I'_G and new sub-domains that must be evaluated for idealization (procedure *Propagate morphology analysis*). Depending on the connection configuration between the two primitives P_i, one of them or both are cut along the contour of I_G to produce the new sub-domains. Then, the MAT is applied to these new sub-domains to update their morphology parameter (procedure *MA morphology analysis*) that reflects the effect of the corresponding merging operation taking place between the two P_i along I_G that stiffens some areas of the two primitives P_i involved. The algorithm terminates when all configurations of C_1 have been processed.

Among the key features of this algorithm, it has to be observed that the influence

[2]Here again like in Section 5.3.1, P_i designates also the set of sub-domains D_{kj} that can result from the decomposition of a primitive P_k when merging it with some other P_l sharing an interface with P_k.

Algorithm 2 Global morphological analysis

```
1: procedure Propagate_morphology_analysis(G_D, x_u)  ▷ The main procedure to extend morphological analyses of sub-domains to the whole object
2:     for each P in list_prims(G_D) do
3:         if P.x > x_u then                                  ▷ If the primitive has to be idealized
4:             for each I_G in list_interfaces_prim(P) do
5:                 P_ngh = Get_connected_primitive(P, I_G)
6:                 if I_G.config = 1 or I_G.config = 4 then
7:                     interVol = Get_interfaceVol(P, P_ngh, I_G)
8:                     P_r = Remove_interfaceVol(P, interVol)   ▷ Update the primitive by removing the volume resulting from interfaces with neighbors
9:                     for i = 1 to Card(P_r) do                ▷ New morphological analysis of the partitions P_r
10:                        P' = Extract_partition(i, P_r)
11:                        P'.x = MA_morpho_analysis(P')
12:                        P_ngh.x = MA_morph_analysis(P_ngh)
13:                        if I_G.config = 1 then
14:                            if P_ngh.x > x_u then
15:                                P_r_ngh = Remove_interVol(P_ngh, interVol)
16:                                interVol.x = MA_morph_analysis(interVol)
17:                                for j = 1 to Card(P_r_ngh) do
18:                                    P'_ngh = Extract_partition(j, P_r_ngh)
19:                                    P'_ngh.x = MA_morpho_analysis(P'_ngh)
20:                                    if interVol.x < x_u then   ▷ If the interVolume is 'idealizable'
21:                                        Merge(P, P_ngh, interVol)   ▷ Merge the intervolume either with P or P_ngh
22:                                    end if
23:                                end for
24:                            else                               ▷ If the interVolume is not 'idealizable'
25:                                P = P'
26:                                Merge(P_ngh, interVol)   ▷ Merge the interVolume with the neighboring primitive which is non 'idealizable'
27:                            end if
28:                            Remove_prim(P_ngh, list_prims(G_D))
29:                        end if
30:                        if P'.x < x_u then                     ▷ if a partition is not 'idealizable'
31:                            Merge(P_ngh, P')   ▷ Merge the partition with the non 'idealizable' primitive neighbor
32:                        end if
33:                    end for
34:                end if
35:            end for
36:        end if
37:    end for
38: end procedure

39: procedure MA_morphology_analysis(P_i) ▷ Procedure using the 2D MAT on the extrusion contour of a primitive
40:     Cont = Get_Contour(P_i)
41:     listofpts = Discretize_contour(Cont)
42:     vor = Voronoi(listofpts)                  ▷ MAT generated using Voronoi diagram of a set of points
43:     maxR = Get_max_radius_of_inscribed_Circles(vor)
44:     x = Set_primitive_idealizableType(maxR, P_i)
45:     return x
46: end procedure
```

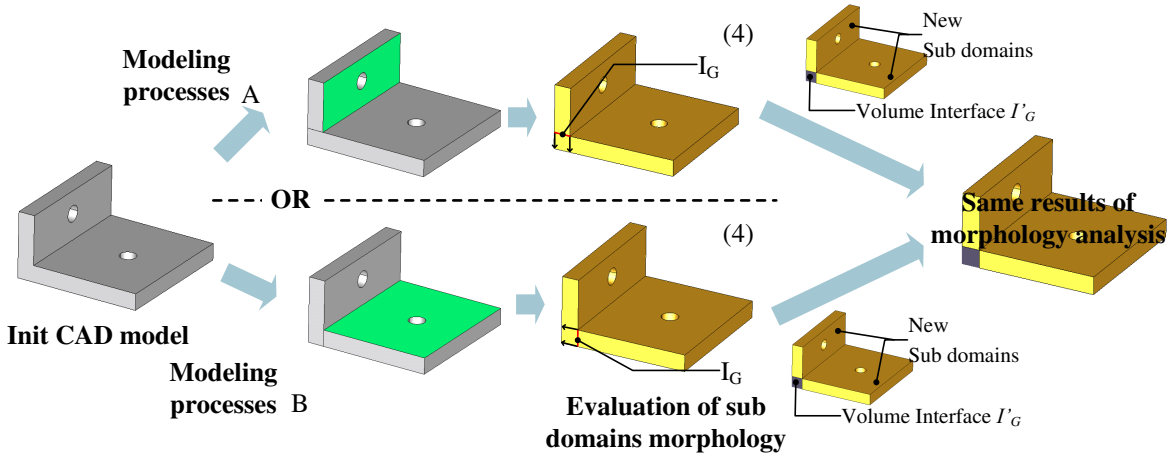


Figure 5.10: Propagation of the morphology analysis of each P_i to the whole object M. A and B illustrates two different sets of primitives decomposing M and numbers in brackets refer to the configuration category of interfaces (see Section 5.3.2.)

of the primitive neighbor P_{ngh} of P_i, is taken into account with the update of P_i that becomes P_r. Indeed, P_r can contain several volume partitions, when $Card(P_r) > 1$, depending on the shapes of P_i and P_{ngh}. Each partition P' of P_r may exhibit a different morphology than that of P_i, which is a more precise idealization indication for the engineer. In case of configuration 1, the overlapping area between P_{ngh} and P_i must be analyzed too, as well as its influence over P_{ngh} that becomes $P_{r_{ngh}}$. Here again, $P_{r_{ngh}}$ may exhibit several partitions, i.e., $Card(P_{r_{ngh}} \geq 1)$, and the morphology of each partition P'_{ngh} must be analyzed. If the common volume of P'_{ngh} and P' is not idealizable, it is merged with either of the stiffest sub-domains P_{ngh} or P_i to preserve the sub-domain the most suited for idealization. In case a partition P' of P_r is not idealizable in configuration 4, this partition can be merged with P_{ngh} if it has a similar morphological status.

Figure 5.10 illustrates this approach with two modeling processes of a simple component. Both processes contain two primitives to be idealized by plate elements and interacting with a surface interface of type (4). The stiffening effect of one primitive on the other creates three sub-domains with interfaces I'_G of type (2). The sub-domain in violet, interacting with both sub-domains to be idealized, can be merged with each of the other sub-domains to create a fully idealized geometry or it can be modeled with a specific connection defined by the user.

Full examples of the extension of the morphological analysis to the whole object M using the interfaces I_G between the primitives of G_D, are given in Figure 5.11. Figures 5.11a, b and c show the sub-domain decomposition obtained after processing the interfaces I_G between primitives P_i of each object M. The same figures illustrate also the update of the morphology criterion on each of these sub-domains when they are iteratively merged through algorithm 2 to form their initial object M. Areas A and B show the stiffening effect of configurations of category (1) on the morphology

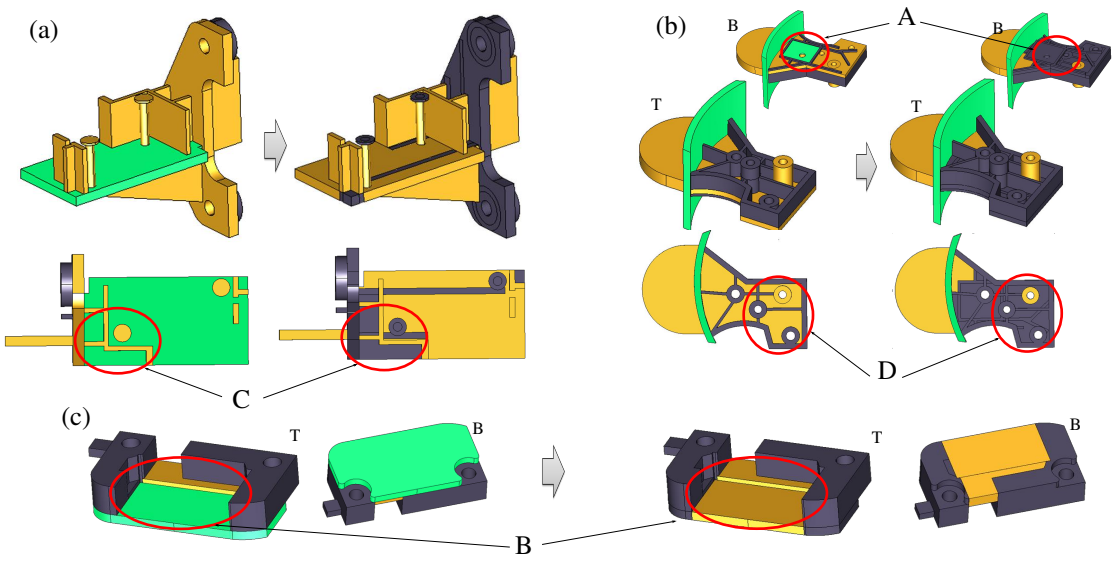


Figure 5.11: Propagation of the morphology analysis on P_i to the whole object M. a, b and c illustrate the influence of the morphology analysis propagation. The analyzed sub-domains are iteratively connected together to form the initial object. T and B indicate the top and bottom views ob the same object, respectively.

of sub-domains of M. Areas C and D are examples of the subdivision produced with configurations of type (4) and the stiffening effects obtained that are characterized by changes in the morphology criterion values.

After applying algorithm 2, one can notice that every sub-domain strictly bounded by one interface I_G of C_2 or by one interface I'_G produced by this algorithm gives a precise idealization information about an area of M. Areas exhibiting connections of type (1) on one or two opposite faces of a sub-domain give also precise information, which is the case for examples of Figure 5.11. However, if there are more piled up configurations of type (1), further analysis is required and will be addressed in the future.

Conclusion

This section has shown how a CAD component can be prepared for idealization. The initial B-Rep geometry has been segmented into a set of extrusion primitives P_i using its construction graph G_D. Using a taxonomy of geometric interfaces, a morphological analysis has been applied to these primitives to identify the 'idealizable' areas over the whole object. As a result, the geometric model is partitioned into volume sub-domains which can be either idealized by shell or beams or not idealized at all. At that stage, only a segmentation of a standalone component has been analyzed. Neither the compatibility with an assembly structure nor the influence of external boundary conditions have been addressed yet, as this is the purpose of the next section.

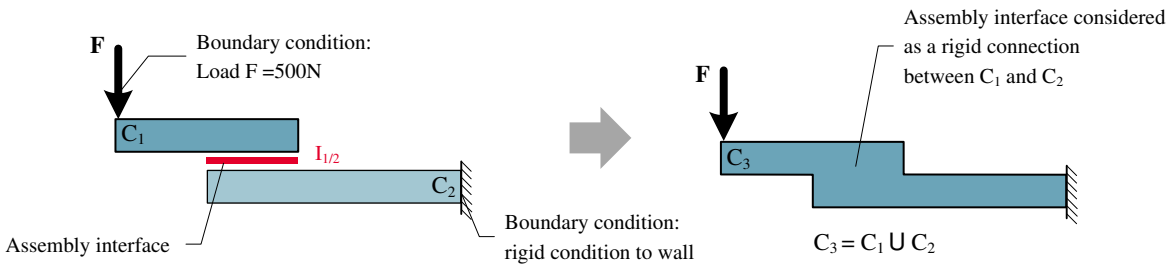


Figure 5.12: Influence of an assembly interface modeling hypothesis over the transformations of two components

5.4 Influence of external boundary conditions and assembly interfaces

As explained in Section 4.7, an assembly model is composed of a set of 3D solid components linked to each other through functional interfaces. A boundary condition that is external to the assembly, as defined in Section 1.4.2, also acts as an interface between the object and its external environment. Figure 5.12 illustrates two types of boundary conditions, a load acting on component C_1 and a rigid connection with the environment on C_2. These areas are defined by the user and are represented as a geometric region on its B-Rep surface which is equivalent to an assembly interface, except that the interface is only represented on one component.

Each component of the assembly can be segmented using respective construction graphs. However, the segmentation of components generates new geometric interfaces between primitives which can be influenced by the assembly interfaces. Therefore, this section aims at studying the role of assembly interfaces and boundary conditions in the idealization process. They can be analyzed either before the morphological analysis as input data or after the segmentation of components.

Impact of the interface modeling hypotheses

Depending on the simulation objectives, the engineer decides if he, resp. she, wants to apply some mechanical behavior over some assembly interfaces (see Table 1.2). This first choice highly influences the components' idealization. As it is highlighted in Section 6.2.1, the engineer may decide not to apply any mechanical behavior at a common interface between two components, e.g., with the definition of rigid connections between their two mesh areas to simulate a continuous medium between components at this interface. This modeling hypothesis at assembly interfaces influences directly the geometric transformations of components. As illustrated in Figure 5.12, a set of components connected by rigid connections can be seen as a unique component after merging them. Therefore, to reduce the complexity of the FEA pre-processing, the

morphological analysis can be applied to this unique component instead of applying it to each component individually.

In case the engineer wants to assign a mechanical behavior to interfaces between components, these interfaces ought to appear in the final idealized model. Now, defining when this assignment can take place during the pre-processing enumerates:

a) at the end of the idealization process, i.e., once the components have been idealized;

b) during the segmentation process, i.e., during the construction graph generation of each component or during their morphological analysis.

These two options are addressed in the following parts of this section. In this section, only the geometric aspect of assembly interfaces is addressed. The transfer of meta-information, e.g., friction coefficient, contact pressure is not discussed here.

Applying assembly interfaces information after components' idealization

In a first step, this part of the section studies the consequences of integrating assembly interfaces at the end of the idealization process, i.e., option (a) mentioned above. These assembly interfaces represent information that is geometrically defined by the interactions between components. These interactions have a physical meaning because the contacts between components exist physically in the final product though a physical contact may not be always represented in a DMU as a common area between the boundaries of two components [SLF*13] (see Section 1.3.2). For sake of simplicity, let us consider that physical contacts are simply mapped to common areas between components. Then, the assembly interfaces are initially prescribed in contrast with geometric interfaces, I_G between primitives of a component that have been created during its segmentation process and aim at facilitating the geometric transformations performed during the idealization process of a component.

One can observe that these assembly interfaces have to be maintained during the dimensional reduction operations of each component. However, these interfaces can hardly be transferred using the only information provided by the idealized models. For example, Figure 5.13b shows that a projection of the assembly interface on the idealized model of C_1 could generate narrow areas which would be difficult to mesh and, on that of C_2, could produce areas that fall outside the idealized model of C_2. This assembly interface has been defined on the initial solid models of C_1 and C_2. If this link is lost during the idealization process, a geometric operator, i.e., like the projection operator just discussed, has to recover this information to adapt this interface on the idealized assembly model. Therefore, to obtain robust transformations of assembly interfaces, these interfaces have to be preserved during the dimensional reduction processes of each

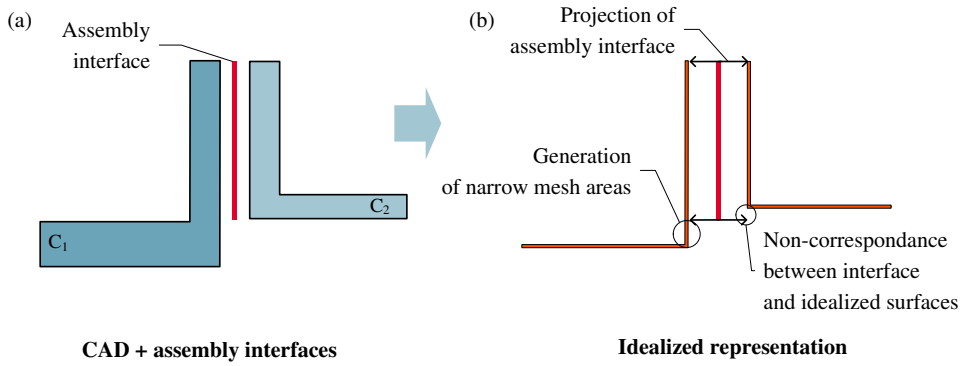


Figure 5.13: Illustration of the inconsistencies between an assembly interface defined between initial CAD components C_1 and C_2 (a) and its projection onto their idealized representations (b).

component. Each of these interfaces, as a portion of the initial solid boundary of a component, has a corresponding representation in its idealized model. This equivalent image would have to be obtained through the transformations applied to the initial solid model of this component to obtain its idealized representation.

Integration of assembly interfaces during the idealization process

In a second step, this part of the section addresses the option (b) mentioned above. As stated in Section 5.4, assembly interfaces have to be generated before the dimensional reduction of assembly components. Now, the purpose is to determine at which stage of the proposed morphological analysis process the interfaces should be integrated. This analysis incorporates the segmentation process, which prepares a component shape for idealization and is dedicated to a standalone 3D solid. This approach can be extended to an assembly model from two perspectives described as follows:

b1) The assembly interfaces and boundary conditions can be used to monitor the definition of specific primitives, e.g., primitives containing the whole assembly interface. Figure 5.14a illustrates such a decomposition with two components C_1 and C_2 fitting the assembly interface with only one primitive in both segmentations. The benefit of this approach lies in avoiding splitting assembly interfaces across various component primitives, which would divide this assembly interface representation across all the primitives boundaries;

b2) The segmentation process is performed independently of the assembly interfaces. Then, they are introduced as additional information when transforming the set of primitives derived from each component and these interfaces are incorporated into the idealized representation of each component. In this case, the intrinsic

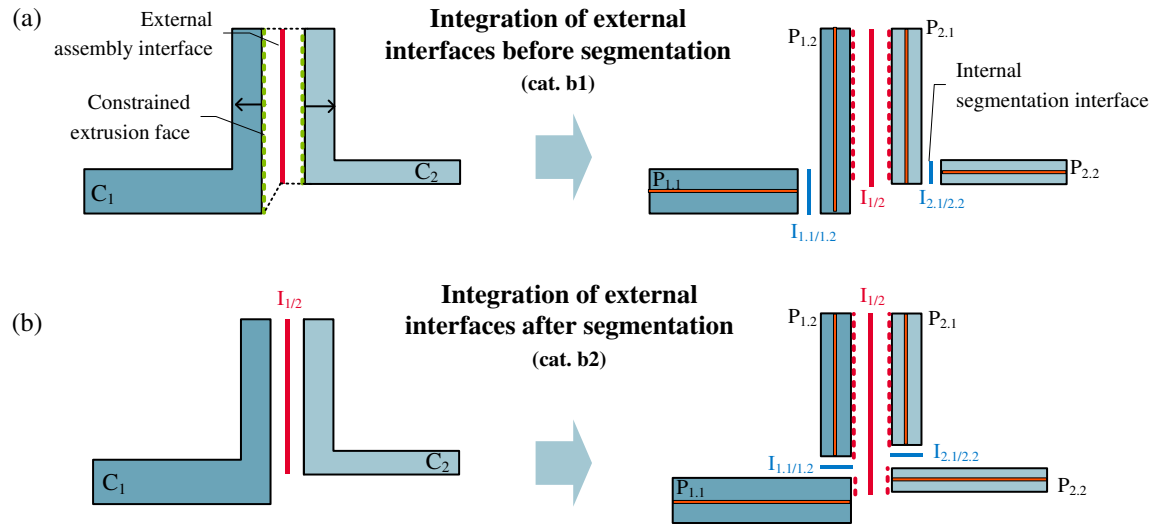


Figure 5.14: Two possible schemes to incorporate assembly interfaces during the segmentation process of components C_1 and C_2: (a) The assembly interface is used to identify extrusion primitives of C_1 and C_2 containing entirely the assembly interface in one extrusion contour, (b) The assembly interface is integrated after the segmentation of components C_1 and C_2 and propagated on each of their primitives.

property of the proposed segmentation approach is preserved and the assembly interfaces are propagated as external parameters on every primitive they are respectively related to. Figure 5.14b shows the imprints of the assembly interface on the primitives derived from the segmentation of components C_1 and C_2.

As a conclusion of the previous analyses, the choice of the idealization process made in this thesis falls into category (b2). Though a fully detailed analysis would bring complementary arguments to each of the previous categories, it appears that category (a) is less robust than category (b), which is an important point when looking for an automation of assembly pre-processing. Within category (b), (b1) leads to a solution that is not intrinsic to the shape of a component, which is the case for (b2). With the current level of analysis, there is no strong argument favoring (b1) over (b2) and (b2) is chosen to keep the level of standalone component pre-preprocessing decoupled from that of the assembly level.

Therefore, assembly interfaces are not constraining the extraction of each the construction graph G_D for each component. During the segmentation process, assembly interfaces are *propagated* only. Rigid connections only are assembly interfaces that can be processed prior the component segmentation process without interfering with it. Indeed, the first part of this section has shown that these interfaces lead to merge the components they connect. Consequently, the rigid interfaces only can be removed from the initial CAD components after the corresponding components have been merged, which simplifies the geometry to be analyzed. From a mechanical point of view, this operation is equivalent to extending the continuum medium describing each compo-

nent because their material parameters and other mechanical parameters are strictly identical.

Then, the other assembly interfaces and external boundary conditions, where a mechanical behavior has to be represented, are propagated through the segmentation process and taken into account during the dimensional reduction process. Chapter 6 carries on with the analysis of interactions between simulation objectives, hypotheses and shape transformations for assembly pre-processing. This helps structuring the preparation process of an assembly in terms of methodology and scope of shape transformation operators. Section 6.4 shows an example of automated idealization of an aeronautical assembly using the idealization process presented in this chapter which is also taken into account in the methodology set in Chapter 6.

Now that the roles of assembly interfaces and external boundary conditions have been clarified, the next section focuses on the dimensional reduction of a set of extrusion primitives connected through geometric interfaces, I_G. The objective is to set up a robust idealization operator enabling the dimensional reduction of extrusion primitives and performing idealized connections between medial surfaces through the analysis of the interface graph G_I of an object M.

5.5 Idealization processes

Having decomposed a assembly component M into extrusion primitives P_i, the last phase of the idealization process consists in the generation and connection of idealized models of each primitive P_i. Now, the interfaces I_G between P_is are precisely identified and can be used to monitor the required deviations regarding medial surfaces. These deviations are needed to improve the idealization process and to take into account the engineer's know-how when preparing a FE model (see discussions of Chapter 2).

Based on the morphological analysis described in Sections 5.2 and 5.3, each P_i has a shape which can be classified in idealization categories of type plate, shell, beam or 3D thick solid. Therefore, depending on P_i's morphological category, a dimensional reduction operator can be used to generate its idealized representation. The geometric model of the idealized P_i is:

- A planar medial surface when P_i has been identified as a plate. This surface corresponds to the extrusion contour offset by half the thickness of this plate along the extrusion direction;

- A medial surface when the primitive has been identified as a shell (see the detailed taxonomy in Tables C.1, C.2). This medial surface is generated as the extrusion of the medial line extracted from the extrusion contour of P_i. This medial line can be generated by applying the 2D MAT to the extrusion contour, as proposed

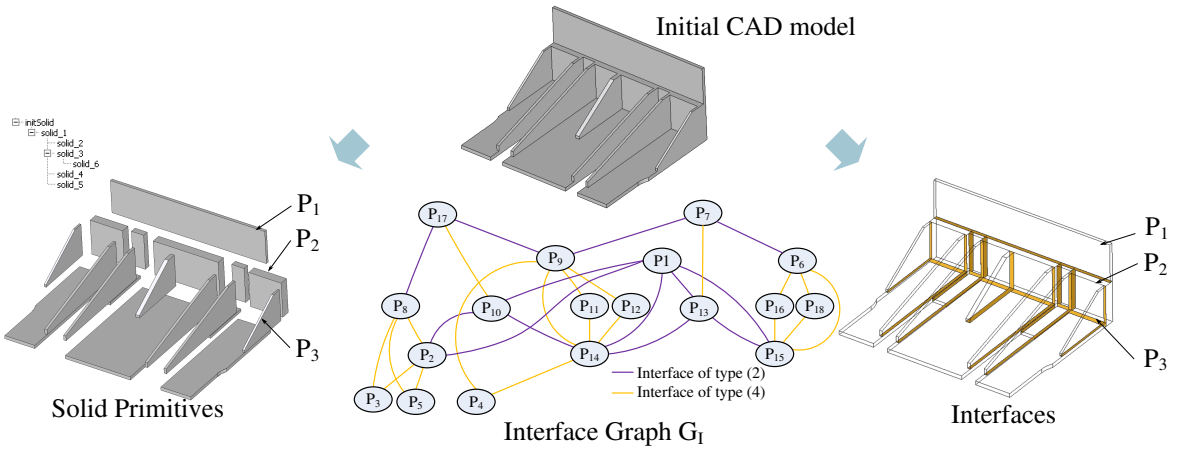


Figure 5.15: Illustration of an interface graph containing I_Gs derived from the segmentation process of M producing G_D.

by Robinson et al. [RAM*06]. The shell thickness varies in accordance with the diameter of circles inscribed in the extrusion contour;

- A medial line when the primitive has been identified as a beam. This line is generated through the extrusion of the point representing the barycenter of the extrusion contour if the beam direction is aligned with the extrusion direction. If the beam direction is orthogonal to the extrusion direction, the medial line corresponds to the medial line of the extrusion contour, offset by half of the extrusion distance;

- The volume domain of P_i when P_i is identified as a 3D thick solid;

since every P_i reduces to an extrusion primitive.

Now that each primitive P_i is idealized individually, the purpose of the following section is to show how the medial surfaces can be robustly connected based on the taxonomy of interfaces I_G illustrated in Figure 5.9.

5.5.1 Linking interfaces to extrusion information

From the construction graph G_D and the geometric interfaces I_G between its primitives P_i, the interface graph G_I can be derived. Figure 5.15 illustrates G_I for a set of extrusion primitives extracted from one component of the aeronautical use-case presented in Figure 1.6.

In G_I, each node, named N_i, is a primitive $P_i \in G_D$ and each arc is a geometric interface I_G between any two primitives P_i and P_j, as I_G has appeared during the segmentation process of M. In a first step, G_I is enriched with the imprints of the

boundary of each I_G on each primitive P_i and P_j that defines this I_G. The j^{th} boundary of an I_G w.r.t. a primitive P_i is noted $C_{j(P_i)}$.

A direct relationship can be established between $C_{j(P_i)}$ and the information related to the extrusion property of P_i. The interface boundary $C_{j(P_i)}$ is classified in accordance with its location over ∂P_i. To this end, each node N_i of G_I representing P_i is subdivided into the subsets: $N_i(F_{b1})$, $N_i(F_{b2})$, $N_i(F_l)$, that designates its base face F_{b1}, its base face F_{b2} and its lateral faces F_l, respectively. Then, $C_{j(P_i)}$ is assigned to the appropriate subset of N_i. As an example, if $C_{j(P_i)}$ has its contours located solely on the base face Fb_1 of P_i, its is assigned to $N_i(F_{b1})$, or if $C_{j(P_i)}$ belongs to one at least of the lateral faces F_l, it is assigned to $N_i(F_l)$. Figure 5.16 illustrates the enrichment of the interface graph G_I of a simple model containing three primitives P_1, P_2 and P_3. For example, the boundary $C_{1(P_1)}$, resulting from the interaction between P_1 and P_3, is assigned to Fb_1 of P_1. Reciprocally, the equivalent $C_{1(P_3)}$ refers to a lateral face of P_3.

The following step determines potential interactions of $C_{j(P_i)}$s over P_i. When a pair of $C_{j(P_i)}$s shares a common geometric area, i.e., their boolean intersection is not null:

$$C_{j(P_i)} \cap C_{k(P_i)} \neq \phi, \tag{5.1}$$

the resulting intersection produces common points or curve segments that are defined as an interface between the pair of interface boundaries $(C_{j(P_i)}, C_{k(P_i)})$ and the nth interface is noted IDn_{C_j/C_k}. Three interfaces between $C_{j(P_i)}$ have been identified on the example of Figure 5.16, e.g., $ID1_{C_1/C_2}$ represents the common edge interaction between $C_{1(P_1)}$ and $C_{2(P_1)}$. These new relations between $C_{j(P_i)}$s form a graph structure G_{ID} where the nodes represent the boundary $C_{j(P_i)}$ and the arcs define the interface IDn_{C_j/C_k}. The graph structure G_{ID} related to a primitive P_i is strictly nested into the i^{th} node of G_I. More globally, the graph structure G_{ID} is nested into G_I.

The graph structures G_{ID} derived from the relations between the boundaries of interfaces I_G of each P_i can be '*merged*' with the interface graph G_I. Let us call G_S this graph (see Figure 5.16d).

5.5.2 Analysis of G_S to generate idealized models

Using G_S, algorithms may be applied to identify specific configurations of connections between idealized primitives. These algorithms are derived from the current industrial practices of idealized FEM generation from B-Rep models. Specific configurations of interface connections can be identified automatically from G_S while allowing the engineer to locally modify the proposed results based on his, resp. her, simulation hypotheses. So, nodes in G_S can be either of type $C_{j(P_i)}$ if there exists a path between $C_{j(P_i)}s$ in P_i or they are P_is if there is no such path. Arcs are built up on either I_Gs or IDn_{C_j/C_k} depending on the type of node derived from $C_{j(P_i)}$ and P_i.

To generate a fully idealized model, i.e., a model where the medial surfaces are

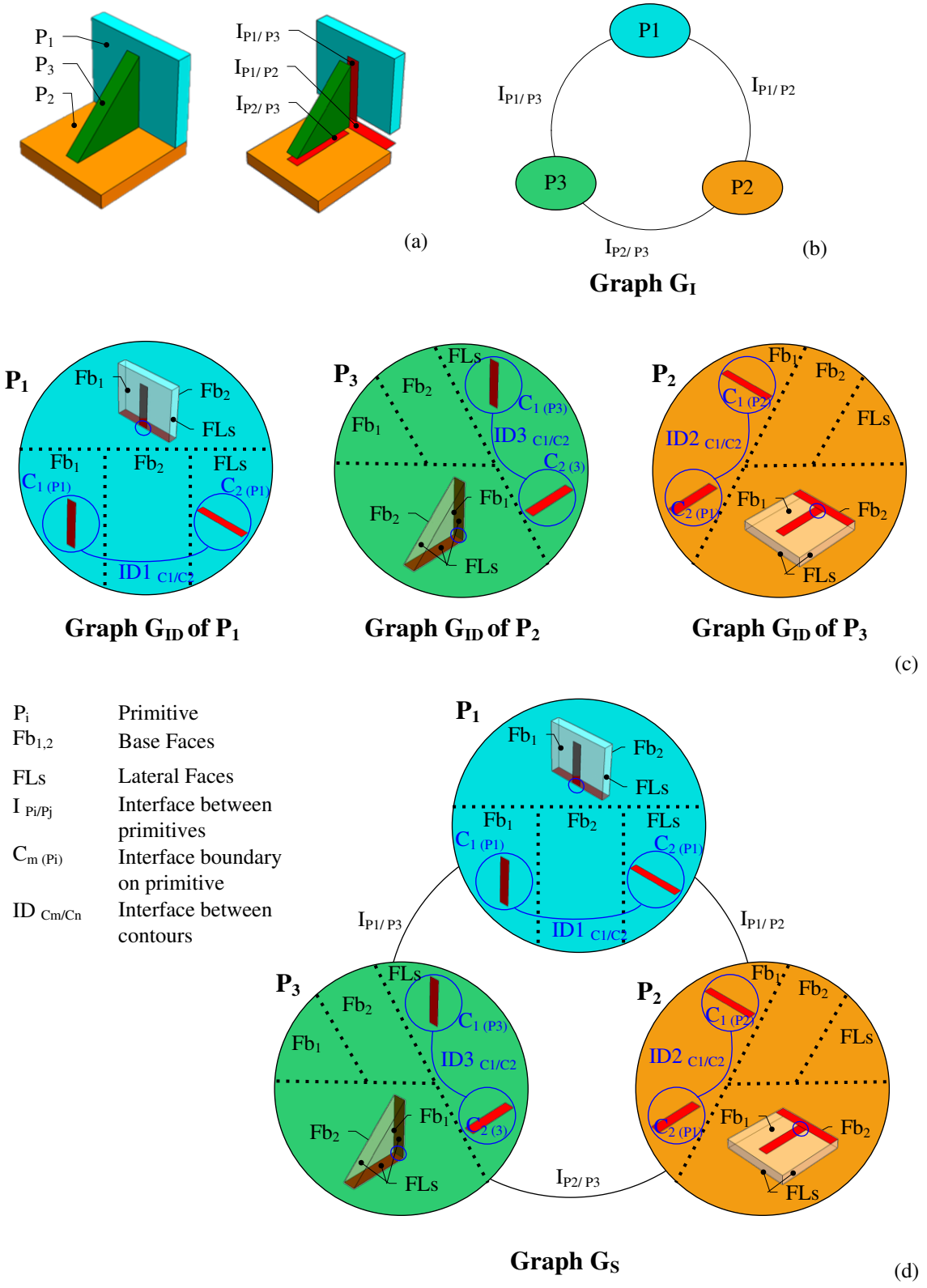


Figure 5.16: Enrichment of the graph G_I with the decomposition of each node into subsets $N_i(F_{b1})$, $N_i(F_{b2})$, $N_i(F_l)$. Illustration of an interface cycle between primitives P_1, P_2 and P_3 built from G_I and G_{ID}. (a) Initial primitives segmentation, (b) G_I graph, (c) G_{ID} for P_1, P_2 and P_3, (d) G_S graph.

connected, three algorithms have been developed to identify respectively:

- interface cycles;
- groups of parallel medial surfaces;
- and L-shaped primitives configurations.

The locations of medial surfaces are described here with orthogonal or parallel properties for sake of simplicity. Therefore, each of them can be generalized to arbitrary angular positions as described in Section 5.3.2. Each algorithm is now briefly described.

Interface cycles

Cycles of interfaces are of particular interest to robustly generate connections among idealized sub-domains. To shorten their description, the focus is placed on a common configuration where all the interfaces between primitives are of type (4). To define a cycle of interfaces of type (4), it is mandatory, in a first step, to identify a cycle in G_I from connections between P_i. In a second step, the structure of connections inside each P_i, as defined in G_{ID}, must contain themselves a path between their interface boundaries $C_{j(P_i)}$s that extends the cycle in G_I to a cycle in $G_S = G_I \cup G_{ID}$. An example of such a cycle is illustrated in Figure 5.16. This level of description of interfaces among sub-domains indicates dependencies between boundaries of medial surfaces. Indeed, such a cycle is a key information to the surface extension operator to connect the set of medial surfaces simultaneously. The medial surfaces perpendicular to their interfaces I_G (of P_3 in Figure 5.16) have to be extended not only to the medial surfaces parallel to their interfaces (of P_1 and P_2 in Figure 5.16), but they have also to be extended in accordance with the extrusion directions of their adjacent primitives. For example, to generate a fully idealized model of the three primitives of Figure 5.16, the corner point of the medial surface of P_3, corresponding to the $ID3_{C_1/C_2}$ edge, has to be extended to intersect the medial surface of P_1 as well as to intersect the medial surface of P_2. As described, the information available in an interface cycle enables a precise and robust generation of connections among idealized sub-domains.

Interface cycles appear as one category of specific idealization processes because they appear frequently in mechanical products and they fall into one category of connection types in the taxonomy of Figure 5.9.

Groups of parallel medial surfaces

Connections of parallel medial surfaces can be handled with medial surface repositioning (see P_1 and P_2 on Figure 5.17a) corresponding to the adjustment of the material thickness on both sides of the idealized surface to generate a mechanical model consistent with the shape of M. This is a current practice in linear analysis that has been advantageously implemented using the relative position of extrusion primitives.

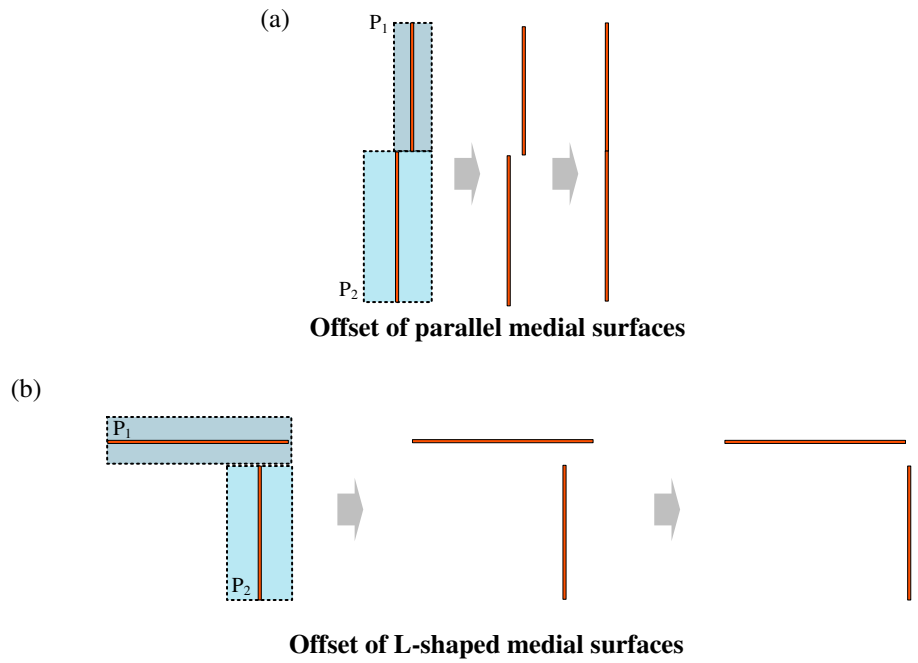


Figure 5.17: Examples of medial surface positioning improvement. (a) Offset of parallel medial surfaces, (b) offset of L-shaped medial surfaces.

These groups of parallel medial surfaces can be identified in the graph G_I as the set of connected paths containing edges of type (2) only. Figure 5.18a shows two groups of parallel medial surfaces extracted from G_I presented in Figure 5.16. As a default processing of these paths, the corresponding parallel medial surfaces are offset to a common average position of the medial surfaces and weighted by their respective areas. However, the user can also snap a medial surface to the outer or inner skins of an extrusion primitive whenever this prescription is compatible with all the primitives involved in the path. Alternatively, he, or she, may even specify a particular offset position. Surfaces are offset to the reference plane as long as the surface remains within the limits of the original volume of the component M. This restriction avoids generating interferences between the set of parallel primitives and the other neighboring primitives. For example, in Figure 5.18, the resulting medial surface of the group of parallel primitives, containing P_1 and P_2, cannot intersect the volumes of its perpendicular primitives such as P_3. This simple process points out the importance to categorize the interfaces between primitives.

Like interface cycles, groups of parallel medial surfaces refer to the taxonomy of Figure 5.9 where they fall into the type (2) category.

L-shaped primitives configurations

When processing an interface of type (4) in G_I, if an interface boundary $C_{j(P_i)}$ is located either on or close to the boundary of the primitive P_j which is parallel to the

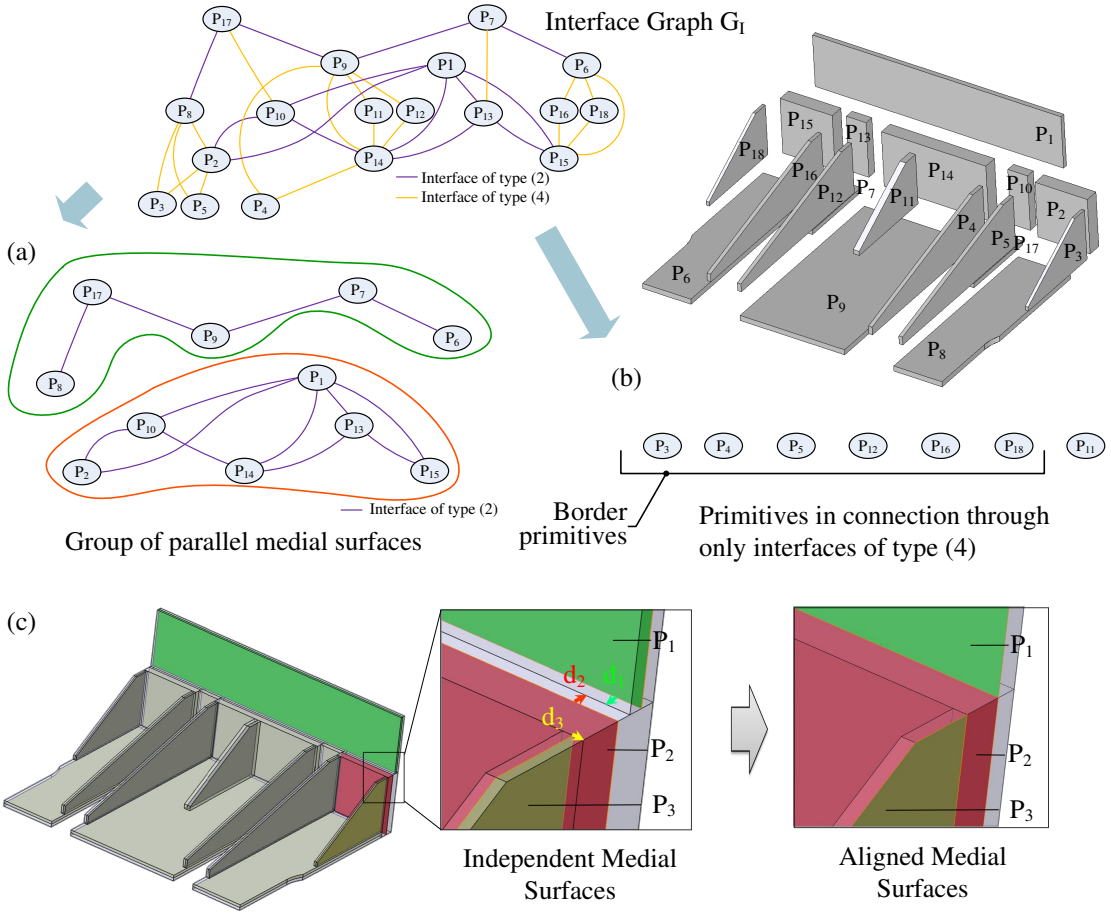


Figure 5.18: Example of identification of a group of parallel medial surfaces and border primitives configurations from the interface graph G_I: (a) extraction of type (2) subgraphs from G_I, (b) set of L-shaped primtives extracted from G_I, (c) initial and idealized configurations of M when groups of parallel primitives and L-shaped configurations have been processed.

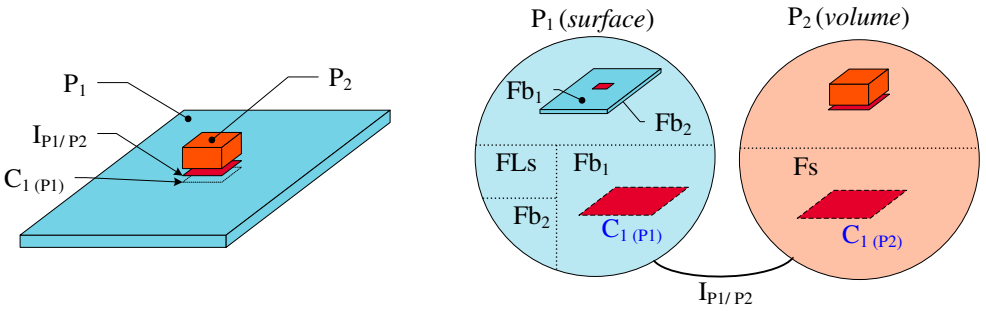


Figure 5.19: Example of a volume detail configuration lying on an idealized primitive.

interface (see P_1 and P_2 on Figure 5.17b or P_2 and P_3 in Figure 5.18c), the medial surfaces needs to be relocated to avoid meshing narrow areas along one of the sub-domain boundaries (here P_3 is moved according to d_3). This relocation is mandatory because $C_{j(P_i)}$ being on or close to the boundary of P_j, the distance between the idealized representation of P_i and the boundary of P_j is of the order of magnitude of the thickness of P_i. Because P_i is idealized, it means that the dimension of FEs is much larger than the thickness of P_i, hence meshing the areas between $C_{j(P_i)}$ and the boundary of P_j would necessarily result in badly shaped FEs. The corresponding configurations are designated as *L-shaped* because P_i and P_j are locally orthogonal or close to orthogonal.

If this configuration refers to mesh generation issues, which have not been addressed yet, L-shaped configurations where a subset of $C_{j(P_i)}$ coincides with the boundary of a connected primitive (see P_2 in Figure 5.18c) can be processed unambiguously without mesh generation parameters, as justified above. Processing configurations where $C_{j(P_i)}$ is only close to a primitive contour requires mesh parameters handling and is left for future work. Primitives connected through interfaces of type (4) only, as illustrated in Figure 5.18b, are part of L-shaped configurations if they have at least a primitive contour $C_{j(P_i)}$ close to P_j boundary. In Figure 5.18b, only P_{11}, which is located in the middle of P_9 and P_{14}, is not considered as an L-shaped primitive. L-shaped configurations can be processed using the precise location of I_G so that the repositioning operated can stay into I_G to ensure the consistency of the idealized model.

Identification criteria of P_i details

The relationships between extrusion information and primitive interfaces may also be combined to analyze more precisely the morphology of standalone primitives, such as small protrusions that can be considered as details. As an example, Figure 5.19 shows the interaction between a primitive P_1, which can be idealized as a plate and a primitive P_2, morphologically not idealizable. The enriched interface graph with G_{ID} indicates that the boundary $C_{1(P_1)}$ lies on a base face, Fb_1, of P_1 whose boundary is used to idealize P_1. Then, if the morphological analysis of Fb_1 is such that: $F = (Fb_1 -^* F_{C_{1(P_1)}})$ shows that F is still idealizable, this means that P_2 has no morphological influence

relatively to P_1, even though P_2 is not idealizable.

As a result, P_2 may be considered as a detail of P_1 and processed accordingly when generating the mesh of P_1, P_2. This simple example illustrates how further analyses can be derived from the graph structures G_{ID} and G_I. Identifying details using the morphological properties of primitives is a way to be independent from the FE mesh size. With the proposed idealization process, a volume can be considered as a detail with respect to the surrounding geometry before the operation of dimensional reduction. This is an a priori approach satisfying the requirement of Section 2.2.1 which stated that a skin detail cannot be directly identified in an idealized model.

Though the criterion described above is not generic, it is illustrative of the ability of the graph structures G_{ID} and G_I to serve as basis of other criteria to cover a much wider range of configurations where skin and topological details could be identified with respect to the idealization processes. A completely structured approach regarding these categories of details is part of future work.

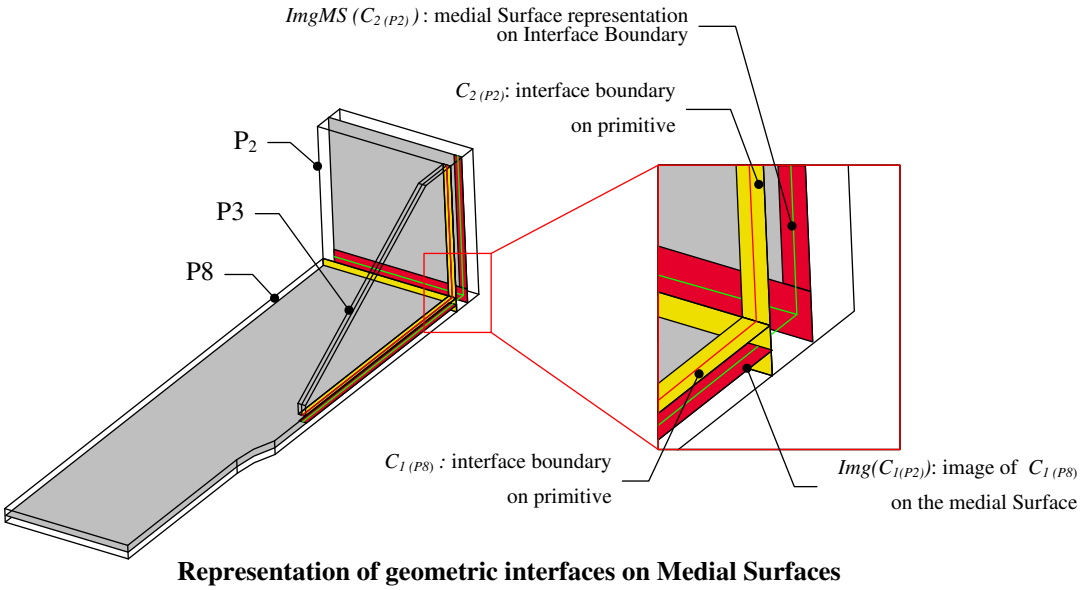
5.5.3 Generation of idealized models

To illustrate the benefits of the interface graph analyses of G_{ID} and G_I, which have been used to identify specific configurations with the algorithms described in Section 5.5.2, an operator has been developed to connect the medial surfaces. Once the groups of parallel medial surfaces have been correctly aligned, the medial surfaces involved in interfaces of type (4) are connected using an extension operator. Because the precise locations of the interfaces between primitives P_i and P_j are known through their geometric imprint $C_{j(P_i)}$ on these primitives, the surface requiring extension is bounded by the imprint of $C_{j(P_i)}$ on the adjacent medial surface. The availability of detailed interface information in G_I and G_{ID} increases the robustness of the connection operator and prevents the generation of inconsistent surfaces located outside interface areas.

Connection operator

Firstly, the connection operator determines the imprints $C_{j(P_i)}$ on the corresponding medial surface of the primitive P_j. This image of $C_{j(P_i)}$ on the medial surface of the neighbor primitive P_i is noted $Img(C_{j(P_i)})$. Figure 5.20a shows, in red, three interface boundaries on the medial surfaces $Img(C_{j(P_i)})$ of the three primitives P_1, P_2, P_3. When adjusting $C_{j(P_i)}$, the medial surface boundary of P_i is also transferred on $Img(C_{j(P_i)})$. Such regions, in green in Figure 5.20a, are noted $ImgMS(C_{j(P_i)})$. The next step extends the medial surfaces involved in interfaces of type (4). The medial surfaces are extended from $Img(C_{j(P_i)})$ to $ImgMS(C_{j(P_i)})$ (the red lines to the green lines in Figure 5.20a). The extensions of the medial surfaces split $Img(C_{j(P_i)})$ into two or more sub-regions. The sub-regions which contains edges coincident with edges of the primitive medial surface are removed to avoid small mesh areas.

(a)



Representation of geometric interfaces on Medial Surfaces

(b)

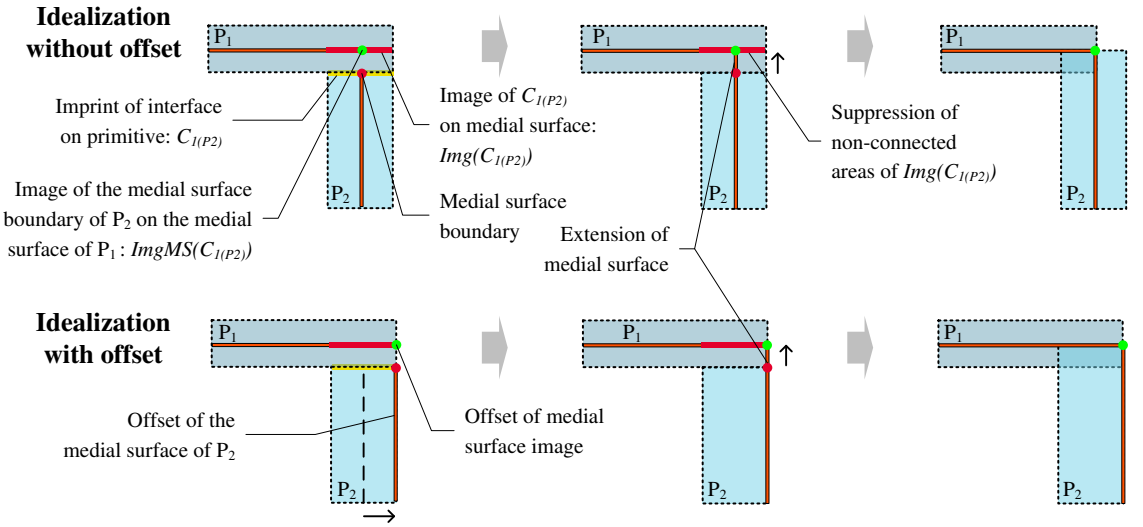


Figure 5.20: (a) Representation of the interface imprints on primitives and on medial surfaces. (b)Connection process of two primitives P_1 and P_2 with and without offsetting their medial surfaces.

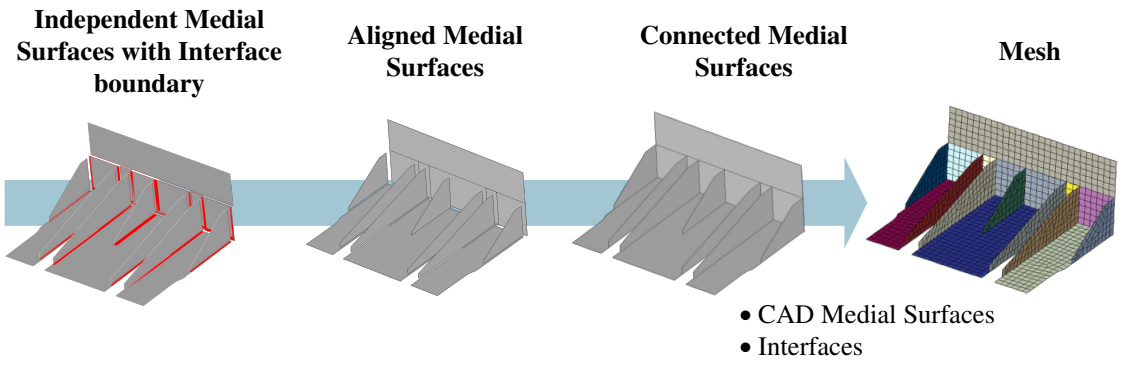


Figure 5.21: Idealization process of a component that takes advantage of its interface graph structures G_{ID} and G_I.

It must be noticed that the regions $ImgMS(C_{j(P_i)})$ lie into P_j and can be obtained easily as a translation of $C_{j(P_i)}$. Therefore, it is not comparable to a general purpose projection operator where existence and uniqueness of a solution is a weakness. Here, $ImgMS(C_{j(P_i)})$ always exists and is uniquely defined from $C_{j(P_i)}$ based on the properties of extrusion primitives.

The interface cycles previously detected are used to identify intersection points within $ImgMS(C_{j(P_i)})$. Figure 5.20a illustrates the interaction between the three $ImgMS(C_{j(P_i)})$ corresponding to the intersection between the three green lines. When processing L-shaped primitives, in case the medial surface of P_i is shifted to the boundary of P_j, the corresponding images $Img(C_{j(P_i)})$ and $ImgMS(C_{j(P_i)})$ are also shifted with the same direction and amplitude. This update of these images preserves the connectivity of the idealized model when extending medial surfaces. Figure 5.20b illustrates the connection process of two primitives P_1 and P_2 with and without moving the medial surface of the primitive P_2. This figure shows how the connection between the idealized representations of P_1 and P_2 can preserve the connectivity of the idealized model.

Results of idealized models

As shown in Figure 5.18b and c, the repositioning of medial surfaces inside P_1, P_2 and P_3 improves their connections and the overall idealized model. Figure 5.21 illustrates the idealization of this component. Firstly, the medial surface of each primitive is generated. Then, the groups of parallel medial surfaces are aligned before the generation of a fully connected idealized model.

Finally, the complete idealization process is illustrated in Figure 5.22. The initial CAD model is segmented using the construction graph generation of Chapter 4 to produce G_D. It produces a set of volume primitives P_i with interfaces between them resulting in the graph structures G_I and G_{ID}. A morphological analysis is applied on each P_i as described in Section 5.3.1. Here, the user has applied a threshold ratio

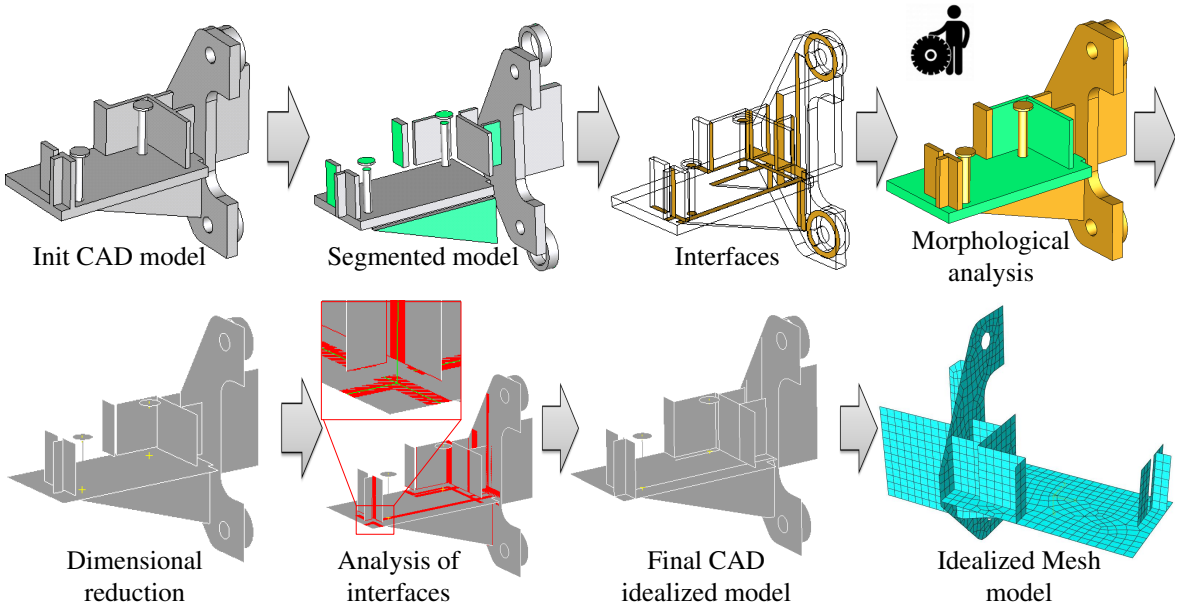


Figure 5.22: Illustration of the successive phases of the idealization process (please read from the left to the right on each of the two rows forming the entire sequence).

$x_u = 2$ and an idealization ratio $x_r = 10$. Using these values, all the primitives are considered to be idealized as surfaces and lines. The final CAD idealized model is generated with the algorithms proposed in Section 5.5.3 and exported to a CAE mesh environment (see Chapter 6).

5.6 Conclusion

In this chapter, an analysis framework dedicated to assembly idealization has been presented. This process exploits construction graphs of components that produce their segmentation into primitives. Morphological criteria have been proposed to evaluate each primitive with respect to their idealization process. The benefits of generative process graphs have been evaluated in the context of idealization processes as needed for FEA.

A morphological analysis forms the basis of an analysis of 'idealizability' of primitives. This analysis takes advantage of geometric interfaces between primitives to assess stiffening effects that potentially propagate across the primitives when they are iteratively merged to regenerate the initial component and to locate idealizable subdomains over this component. Although the idealization concentrates on shell and plates, it has been observed that the morphological analysis can be extended to derive beam idealizations from primitives.

This morphological analysis also supports the characterization of geometric details in relation to local and to idealizable regions of a component, independently of any nu-

merical method used to compute solution fields. Overall, the construction graph allows an engineer to access non trivial variants of the shape decomposition into primitives, which can be useful to evaluate different idealizations of a component.

Finally, this decomposition produces an accurate description into sub-domains and into geometric interfaces which can be used to apply dimensional reduction operators. These operators are effectively robust because interfaces between primitives are precisely defined and they combine with the primitives to bound their idealized representations and monitor the connections of the idealized model.

The principle of component segmentation appears also to be compatible with the more general needs to process assembly models. Indeed, components are sub-domains of assemblies and interfaces are also required explicitly to be able to let the engineer assign them specific mechanical behavior as needed to meet the simulation objectives.

The proposed idealization process can now take part to the methodology dedicated to the adaption of a DMU to FE assembly models, as described in the next chapter.

Chapter 6

Toward a methodology to adapt an enriched DMU to FE assembly models

Having detailed the idealization process as a high-level operator taking benefits from a robust shape enrichment, this chapter extends the approach toward a methodology to adapt an enriched DMU to FE assembly models. Shape transformations resulting from user-specified hypotheses are analyzed to extract preprocessing tasks dependencies. These dependencies lead to the specification of a model preparation methodology that addresses the shape transformation categories specific to assemblies. To prove the efficiency of the proposed methodology, corresponding operators have been developed and applied to an industrial DMU. The obtained results point out a reduction in preparation times compared to purely interactive processes. This time saved enables the automation of simulation processes of large assemblies.

6.1 Introduction

Chapter 3 set the objectives of a new approach to efficiently adapt CAD assembly models derived from DMUs as required for FE assembly models. Chapters 4 and 5 significantly contributed to solve two issues regarding the proposed approach. The first challenge addresses the internal structure of CAD components that has to be improved to provide the engineer with a robust segmentation that can be used as basis for a morphological analysis. The second challenge deals with the implementation of a robust idealization process automating the tedious tasks of dimensional reduction operations and particularly the treatment of connections between idealized areas. Then, the proposed algorithms have been specified to enable the transformations of solid primitives as well as their associated interfaces. The set of solid primitives can result either from a component segmentation or an assembly structure decomposed

into components in turn decomposed into solid primitives. Thus, the method allows an engineer to transform components' shapes while integrating the semantics of assembly interfaces.

This chapter goes even further to widen the scope of shape transformations at the assembly level and evolve toward a methodology of assembly pre-processing. The aim is to enforce the ability of the proposed approach to challenge the current practices to generate large assembly simulation models. The analysis of dependencies among component shape transformations applied to assemblies will help us to formalize this methodology. Thanks to the geometric interfaces of components and functional information expressed as functional designations of components obtained with the method of Shahwan et al. [SLF*13] summarized in Section 3.3, new enriched DMU are now available to engineers. Thanks also to the component segmentation into solid primitives and their interfaces that can be used to idealize sub-domains as described in Chapter 5, the models input to FEA pre-processing contains much more information available to automate the geometric transformations required to meet the simulation objectives. The method described in Section 6.1 of this chapter uses this enriched DMU as input to structure the interactions between shape transformations, leading to a methodology which structures the assembly preparation process. To prove the validity of the proposed methodology, Sections 6.3 and 6.4 illustrate it with two test cases of an industrial assembly structure (see Figure 1.6) to create a simplified volume model and an idealized surface model. To this end, new developments are presented that are based on operators that perform shape transformations using functional information to efficiently automate the pre-processing. Section 6.3 develops the concept of template-based transformation operators to efficiently transform groups of components. This operator is illustratively applied to an industrial aeronautical use-case with transformations of bolted junctions. Section 6.4 deploys the methodology using the idealization algorithms of Chapter 5 to generate a fully idealized assembly model. Finally, the software platform developed in this thesis is presented at the end of Section 6.4.

6.2 A general methodology to assembly adaptions for FEA

Chapter 1 pointed out the impact of interactions between components on assembly transformation. The idealization of components is not the only time consuming task during assembly preparation. When setting up large structural assembly simulations, processing contacts between components as well as transforming entire groups of components are also tedious tasks for the engineer. The conclusion of Section 1.5.4 showed that the shape transformations taking place during an assembly simulation preparation process interact with simulation objectives, hypotheses and functions attached to components and to their interfaces. Therefore, to reduce the amount of time spent

on assembly pre-processing, the purpose is now to analyze and structure the interactions between shape transformations. This leads to a methodology that structures the assembly preparation process.

6.2.1 From simulation objectives to shape transformations

How do shape transformations emerge from simulation objectives and how do they interact between themselves? This is to be analyzed in the following section. However, the intention is not to detail interactions but to focus on issues that help to structure the shape transformations. Transformation criteria related to time that may influence simulation objectives are not relevant, i.e., manual operations that have been performed to save time are irrelevant. Indeed, the purpose is to structure shape transformations to save time and improve the efficiency of preparation processes.

6.2.1.1 Observation areas

From the simulation objectives, the structural engineer derives hypotheses that address components and/or interfaces among them, hence the concept of observation area.

Even if this engineer has to produce an efficient simplified model of the assembly to meet performance requirements, anyhow he/she must be able to claim that his/her result is correct and accurate enough in critical observations areas that are consistent with the simulation objectives. Therefore, the mechanical model set up in these areas must remain as close as possible to the real behavior of the assembly. Thus, the geometric transformations performed in these areas must be addressed in a first place. As an example, in Figure 6.1, the simulation objective is to observe displacements in the identified region (circled area) due to the effects of local loading configurations, the section of the domain being complex. A possible engineers hypothesis can be to model precisely the 3D deformation in the observation area with a volume model and a fine mesh and set up a coarse mesh or even idealized sub-domains outside the area of interest. To explicit this hypothesis over the domain, the circled area should be delimited before meshing the whole object. During a preparation process, setting up observation areas and thus, subdividing an assembly into sub-domains, independently of the component boundaries and their interfaces, acts as a prominent task.

6.2.1.2 Entire idealization of components

Idealizations have inherently a strong impact on shape transformations because of their dimensional reduction. Applied to a standalone component, idealization is meaningful to transform 3D domains up to 1D ones. In the context of assemblies, to meet simulation objectives, performances, and reduce the number of unknowns, the engineer

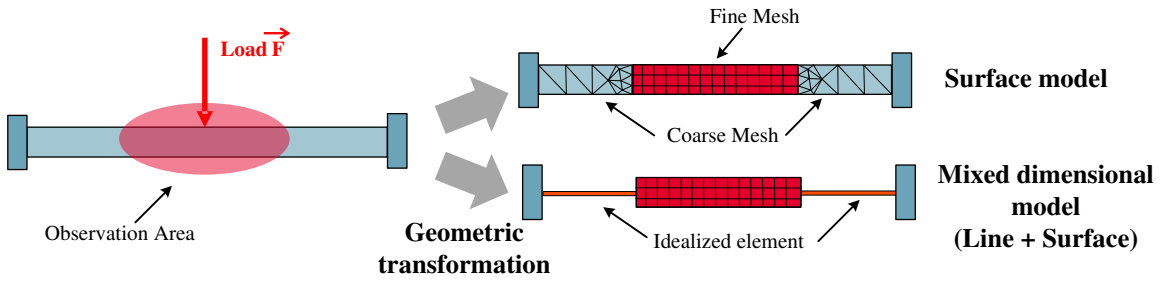


Figure 6.1: Setting up an observation area consistent with simulation objectives.

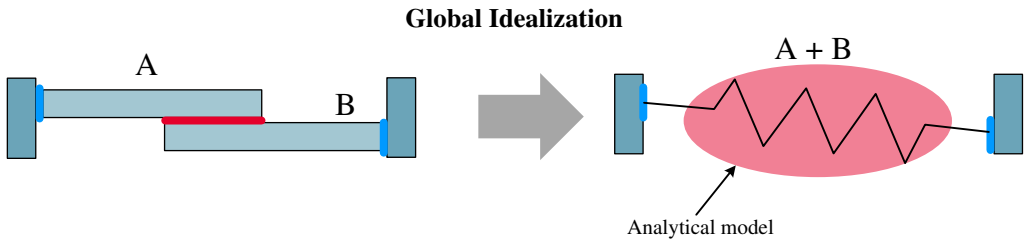


Figure 6.2: Entire idealization of two components.

can idealize a component up to a point (0D), e.g., a concentrated mass, or even replace it by a pre-defined solution field, e.g., a rigid body behavior or a spring-damper field. When analytical models are available, some groups of components, like the bolts in Figure 6.3a, do not appear geometrically in the FE assembly. The planar flange connected by the bolts forming the major interface is used as location of a section in the FE assembly model to determine resulting internal forces and moments in that section. Then, the analytical model is independent of the FE one and it is fed with these parameters to determine the pre-stress parameters of the bolts. Figure 6.3b illustrates the complete idealization of pulleys as boundary conditions. This time, an analytical model has been used prior to the FE assembly model. Such categories of idealizations can be also applied to a set of connected components (see Figure 6.2). In either case, such transformations have a strong impact on the interfaces between the idealized components and their neighboring ones.

Consequently, interfaces between idealized components can no longer be subjected to other hypotheses, e.g., contact and/or friction. Again, this observation highlights the prominence of idealization transformations over interfaces ones.

6.2.1.3 Processing Interfaces

Interfaces between components are the location of specific hypotheses (see Table 1.2) since they characterize junctions between components. Naturally, they interact with hypotheses and shape transformations applied to the components they connect. Let

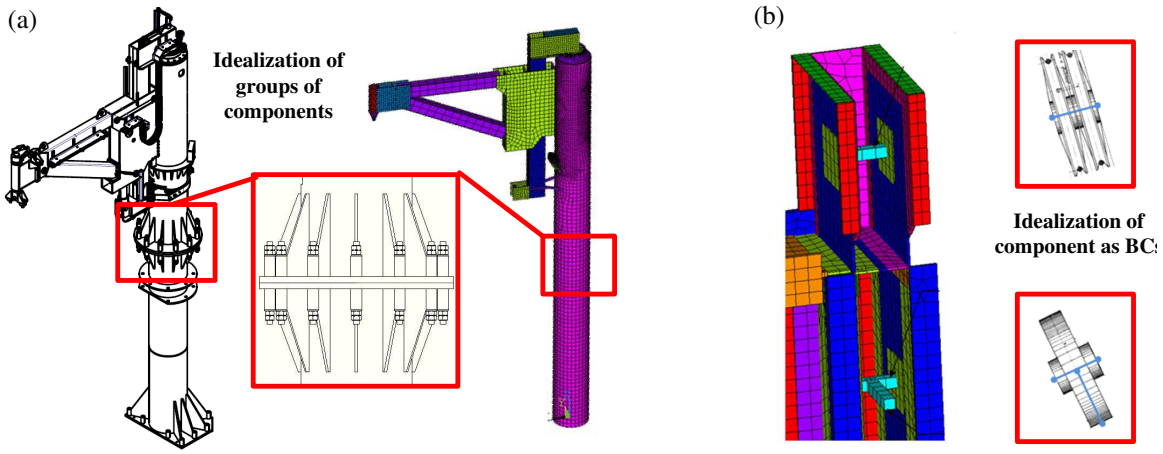


Figure 6.3: (a) Transformation of groups of components as analytical models and, (b) idealization of components as BCs (courtesy of ANTECIM).

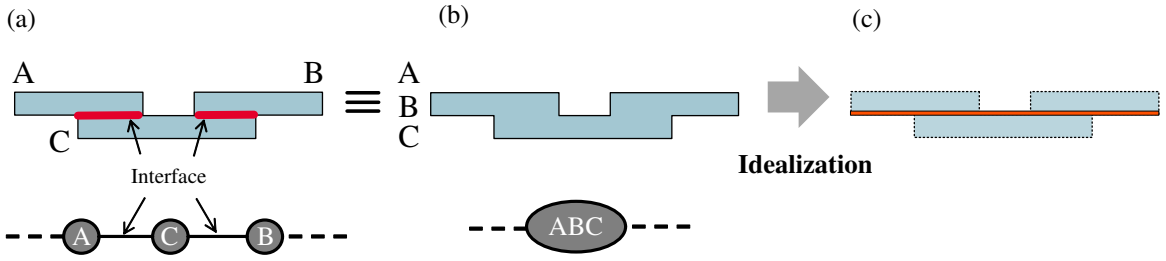


Figure 6.4: Influence of interfaces over shape transformations of components.

us consider the example of Figure 6.4. In a first place, a simulation objective can be stated as: modeling the deformation of the assembly with relative movements of plates A, B, C under friction. Under this objective, hypotheses are derived that require modeling interfaces (A, C) and (B, C) with contact and friction. Then, even if A, B and C, as standalone components, can be candidate to idealization transformations, these idealizations cannot be idealized further because the interfaces would need to be removed, which is incompatible with the hypotheses. In a second place, another simulation objective can be stated as: modeling the deformation of the assembly where the junctions between plates A, B, C are perfect, i.e., they behave like a continuous medium. There, plates A, B, C can still be idealized as standalone components but the hypothesis on interfaces enables merging the three domains (Figure 6.4b) and idealizing further the components to obtain an even simpler model with variable thickness (see Figure 6.4c).

Thus, there are priorities between shape transformations deriving from the hypotheses applied to interfaces. Indeed, this indicates that hypotheses and shape transformations addressing the interfaces should take place before those addressing components as standalone objects. Effectively, interfaces are part of component boundaries; hence

their transformations modify these boundaries. It is more efficient to evolve the shape of interfaces alone first and to process component shapes, as isolated domains, afterwards. As explained in Section 5.4, once the role of interfaces has been defined in the assembly according to the user's simulation objectives and the corresponding transformations have been performed, each individual component can be transformed on its own to take into account these interfaces as external boundary conditions during its idealization/simplification process.

6.2.2 Structuring dependencies between shape transformations as contribution to a methodology of assembly preparation

Section 6.2.1 has analyzed the relationships between simulation objectives, hypotheses, and shape transformations of assemblies. One outcome of this section structures the dependencies between hypotheses and shape transformations that address an assembly at different levels. The purpose is now to exploit these dependencies to organize the various steps of an assembly simulation preparation process so that it appears as linear as possible to be efficiently automatized.

Dependencies of geometric transformations of components and interfaces upon simulation hypotheses

Section 6.2.1.1 has shown the dependency of observation areas upon the simulation objectives. Defining observation areas acts as a partitioning operation of an assembly, independently of its components boundaries. Section 6.2.1.2 introduced the concept of entire idealization of components and pre-defined solutions fields. Indeed, the shape transformations derived from Section 6.2.1.2 cover also sub-domains over the assembly that can be designated as '*areas of weak interest*'. There, the assembly interfaces contained in these areas are superseded by the transformations of Section 6.2.1.2. From a complementary point of view, areas of interest, once defined, contain sub-domains, i.e., components or parts of components, that can still be subjected to idealizations, especially transformations of volumes sub-domains into shells/membranes and/or plates. Consequently, areas of weak interest are regarded as primary sub-domains to be defined. Then, entire idealization of components and pre-defined solutions fields will take place inside these areas, in a first place (identified as **task 1** in Figure 6.5). These areas are necessarily disjoint from the areas of interest, therefore their processing cannot interfere with that of areas of interest.

Sections 1.5.4 and 6.2.1.3 have shown that hypotheses about assembly interfaces influence the transformations of component boundaries. Hence, these hypotheses must be located outside of areas of weak interest to preserve the consistency of the overall simulation model. Subsequently, these hypotheses about interfaces are known once the

areas of weak interest have been specified. Consequently, they come as a **task 2**, after the definition of the areas of weak interest and the corresponding shape transformations of assembly interfaces should be applied at that stage.

As highlighted at Sections 1.5.3, 1.5.4, and 6.2.1.3, idealizations are shape transformations having an important impact on component shapes. As mentioned at Section 2.2, the order of detail removal operations and idealizations has not been studied precisely yet. However, once idealizations have been assigned to sub-domains corresponding to primitives P_i of components, these transformations produce new interfaces between these sub-domains (see Figure 5.1) in addition to the assembly interfaces originated from the interactions between components. Independently of skin and topological details, idealizations can be regarded as **task 3** in the preparation process flow. Effectively, these new interfaces are the consequences of idealizations of sub-domains that result from idealization processes. Therefore, these new interfaces cannot be processed during the second task. These new interfaces should be processed in a first place after the idealizations performed during the third task. The corresponding shape transformations attached to these new interfaces form **task 4**.

Now, as pointed out at Section 6.2.1.3, idealizations can interact between themselves because the idealized sub-domains can be extended/merged in accordance to their geometric configurations to produce a connected idealized model wherever it is required by the simulation objectives. This new set of shape transformations can be regarded as **task 5** that could indeed appear as part of an iterative process spanning tasks three and four. This has not yet been deeply addressed to characterize further these stages and conclude about a really iterative process or not. Even though task two addresses hypotheses attached to assembly interfaces and their corresponding shape transformations, it cannot be swapped with task three to contribute to iterative processes discussed before. Indeed, task 2 is connected to assembly interfaces between components and their processing could be influenced by component idealizations, e.g., in a shaft/bearing junction, idealizing the shaft influences its contact area with the bearings that guide its rotational movement.

Hypotheses and shape transformations previously mentioned enable the definition of a mechanical model over each sub-domain resulting from the tasks described above but this model must be available among the entities of CAE software. This is mandatory to take advantage of this software where the FE mesh will be generated. Consequently, if an engineer defines interface transformations consistent with the simulation hypotheses, there may be further restrictions to ensure that the shapes and mechanical models produced are effectively compatible with the targeted CAE software capabilities. For sake of conciseness, this aspect is not addressed here.

Toward a methodology of assembly model preparation

This section has identified dependencies among shape transformations connected to

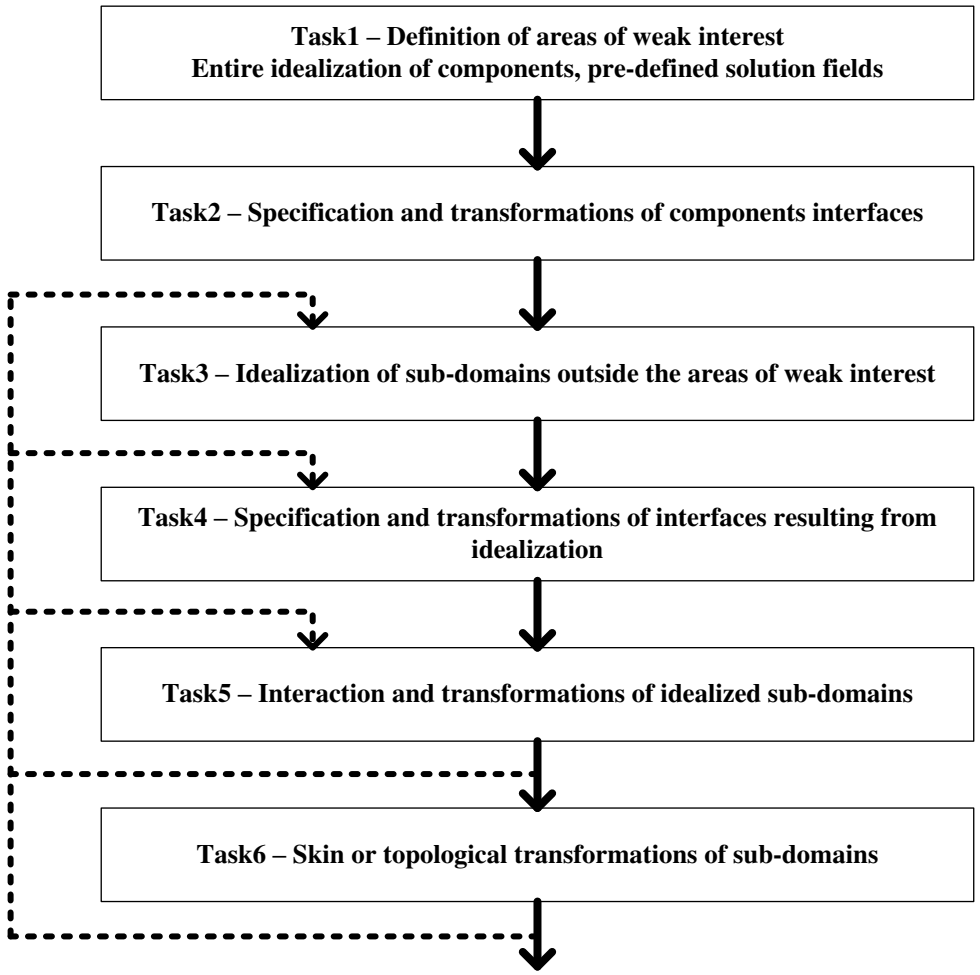


Figure 6.5: Synthesis of the structure of an assembly simulation preparation process.

simulation objectives and hypotheses. Shape details on components can be identified using the morphological analysis, as illustrated in Section 5.5.2. This analysis has shown that the primitives P_i obtained from the construction graph G_D could be further decomposed into sub-domains after analyzing the result of a first MAT. This analysis has also shown its ability to categorize sub-domains relatively to each other. However, detail removal, which originates from different components or even represents an entire component, needs to be identified through a full morphological analysis of the assembly. This has not been investigated further and is part of future research. Currently, detail removals can take place after task two but they can be prior or posterior to idealizations. The definition of areas of interest has connections with the mesh generation process to monitor the level of discretization of sub-domains. This definition acts as a partitioning process that can take place at any time during the process flow of Figure 6.5.

6.2.3 Conclusion and methodology implementation

As a conclusion, the difference between a simulation model of a standalone component and that of an assembly relates to:

- The interactions between components. The engineer formulates a hypothesis for each interface between components. These hypotheses derive from assembly simulation objectives;

- The ordering of shape transformations. The entire idealization of components and the specification of pre-defined solution fields followed by shape transformations of component interfaces are prioritized;

- The interactions between idealizations and assembly interface transformations. To be able to model large assemblies, not only components but groups of components have to be idealized, which can significantly increase the amount of interactions between idealizations and transformations of assembly interfaces.

The simulation objectives are expressed through hypotheses that trigger shape transformations. Studying the interactions between simulation objectives, hypotheses, and shape transformations has revealed dependencies between categories of shape transformations. These dependencies have been organized to structure the assembly simulation model preparation process in terms of methodology and scope of shape transformation operators. The proposed methodology aims at successfully selecting and applying the geometric transformation operators corresponding to the simulation objectives of the engineer.

Starting from an enriched structure of DMU as proposed in Section 3.3, the purpose of the next sections is to illustrate how this methodology can be applied to industrial use-cases. Two implementations are proposed and both are based on the exploitation of functional features of the assembly using the interfaces between components (see Figures 2.14 and 3.2).

As a first methodology implementation, Section 6.3 develops the concept of template-based operators. This concept uses functional information and the geometry of assembly interfaces to identify configurations such as bolted junctions and to apply specific simulation hypotheses to transform their assembly interfaces. This transformation creates a simplified volume model with a sub-domain decomposition around bolted junctions, as required by the simulation objectives (see Figure 6.6).

The second methodology implementation, presented in Section 6.4, leads to a full idealization of an assembly use-case. This implementation confirms that the idealization process of Chapter 5 can be generalized to assembly structures.

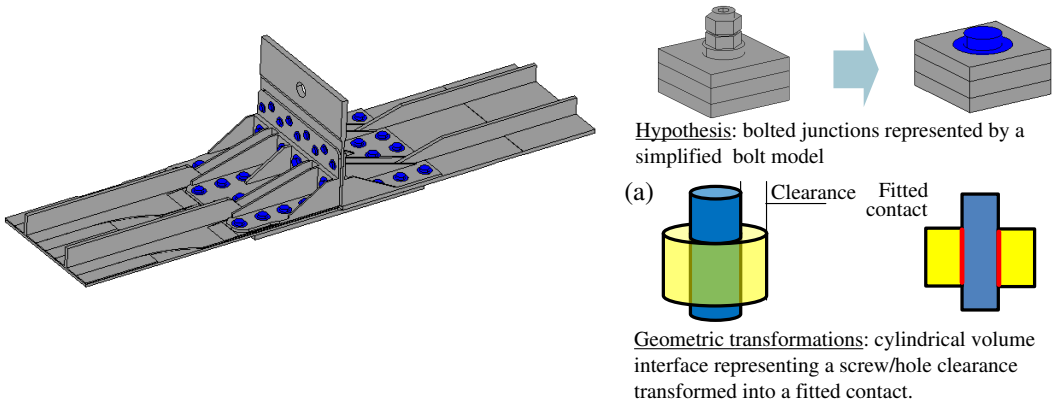


Figure 6.6: Use-Case 1: simplified solid model with sub-domains decomposition around bolted junctions. Area enlarged (a): Illustration of task 2 that transforms bolted junction interfaces into fitted contacts.

6.3 Template-based geometric transformations resulting from function identifications

As illustrated in Section 1.5.4, repetitive configurations, e.g., junctions, and their processing are critical when preparing assembly structures, justifying the need to automate the preparation of large assembly models. To improve the efficiency of DMU transformations for FEA, Section 3.5.2 has proposed to set up relationships between simulation objectives and geometric transformations through the symbolic representation of component functions and component interfaces. The method is based on an enriched DMU as input (see Section 3.3) which contains explicit geometric interfaces between components (contacts and interferences) as well as their functional designations. This enriched DMU has been generated based on the research work of Shahwan et al. [SLF*12, SLF*13]. The geometric interfaces feed instances of conventional interfaces (CI) (see Section 1.3.2) classes structured into a taxonomy, $\mathcal{T}_{CI}$, that binds geometric and symbolic data, e.g. planar contact, spherical partial contact, cylindrical interference, ... Simultaneously, CI and assembly components are organized into a CI graph: CIG(C, I) where the components C are nodes and CI are arcs.

Starting from this enriched model, Section 6.3.2 extends the functional structure to reach a level of product functions. Therefore, simulation objectives can be used to specify new geometric operators using these functions to robustly identify the components and assembly interfaces to transform [BSL*14]. If close to Knowledge Based Engineering (KBE), this scheme is nonetheless more generic and more robust than KBE approaches due to the fact that functional designations and functions are generic concepts. KBE aims at structuring engineering knowledge and at processing it with symbolic representations [CP99, Roc12] using language-based approaches. Here, the focus is on a robust connection between geometric models and symbolic representations

featuring functions.

To prove the validity of this approach and of the methodology proposed in Section 6.2.2, this section presents a template-based operator dedicated to the automation of shape transformations of bolted junctions (see Figure 6.6). The template operator is described in Section 6.3.3. Using functional information and geometric interfaces, this operator applies a user-defined template to simplify bolts and sets control sub-domains around them in their associated tightened components to enable the description of the friction phenomenon between these components. This template is used to precisely monitor the mesh generation process while preserving the consistency of contacts and adapting the assembly model to simulation objectives. Finally, Section 6.3.4 illustrates the result of the different tasks of the proposed methodology applied to the transformation of the bolted junctions of the root joint model presented in Figure 1.6.

6.3.1 Overview of the template-based process

The overall principle of the template-based approach is introduced in Figure 6.7. It uses the available functional information and geometric interfaces, see (1) in Figure 6.7, as well as a library of pre-defined parametric templates, see (2). From this library, the templates are selected by the user according to his, resp. her, simulation objectives. Once the templates have been selected, the operator automatically identifies the functions in the assembly the templates are related to, see (3). Then, as explained in Section 6.3.2, the operator identifies in the CAD assembly the components and interfaces to be transformed, see (4). In (5), the templates definition are fitted to the real geometry, i.e. the components and interfaces dimensions involved in the geometrical transformations are updated in the pre-definition of the templates. Section 6.3.3.1 detailed the compatibility conditions required by the templates insertion in the real geometry. Finally, the real geoemtry is transformed according to the compatibility conditions and the templates are inserted in the assembly model, see task 6. Section 6.3.4 describes the application of template operator on two aeronautical use-cases. It results in a new CAD assembly model adapted to simulation objectives.

6.3.2 From component functional designation of an enriched DMU to product functions

Though the bottom-up approach of Shahwan et al. [SLF*12, SLF*13] summarized in Section 3.3 provides assembly components with a structured model incorporating functional information that is independent of their dimensions, their functional designation does not appear as an appropriate entry point to derive shape transformation operators as required for FE analyses. Indeed, to set up FE assembly models, an engineer looks for bolted junctions that he, resp. she, wants to transform to express friction phenomena, pre-stressed state in the screw, ... Consequently, the functional level needed is not

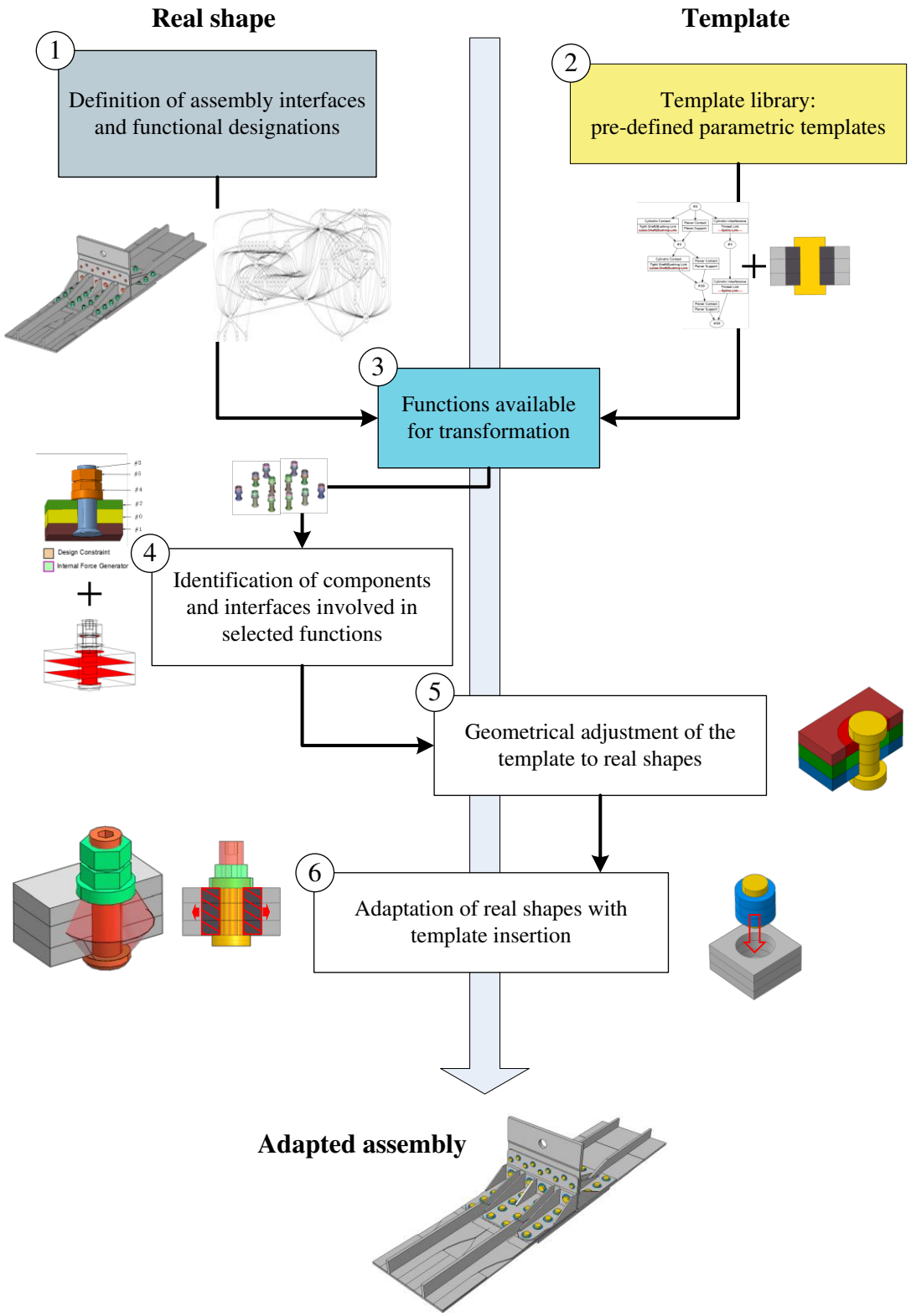


Figure 6.7: Overview of the main phases of the template-based process.

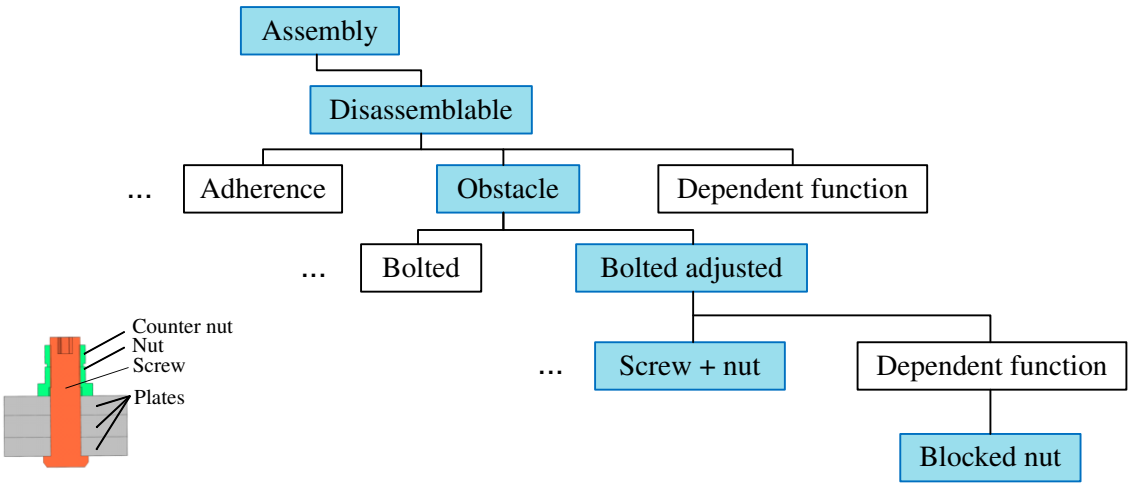


Figure 6.8: Subset of $\mathcal{T}_{FN}$, defining a functional structure of an assembly.

the functional designation, which is bound to a single component, it is the product function itself that is needed to address the corresponding set of components and their functional interfaces.

To this end, it is mandatory to refer to product functions. This is achieved with a taxonomy of functions, $\mathcal{T}_{FN}$, that can produce a functional structure of an assembly (see Figure 6.8). Blue items define the sub-path in $\mathcal{T}_{FN}$ hierarchy that characterizes bolted junctions.

Each instance of a class in $\mathcal{T}_{FN}$ contains a set of components identified by their functional designation, i.e., it contains their structured geometric models and functional interfaces. As a result of the use of $\mathcal{T}_{FN}$, a component of a DMU can be automatically identified when it falls into the category of cap screws, nuts, locking nuts, that are required to define bolted junctions. This means that their B-Rep model incorporates their geometric interfaces with neighboring components. The graph of assembly interfaces set up as input to the process of functional designation assignment, identifies the components contained in a bolted junction. Each component is assigned a functional designation that intrinsically identifies cap screws, nuts, locking nuts, ..., and connects it with an assembly instance in $\mathcal{T}_{FN}$.

It is now the purpose of Section 6.3 to take advantage of this information to set up the template-based transformations.

6.3.3 Exploitation of Template-based approach for FE models transformations

As a result of the functional enrichment process, the DMU is now geometrically structured, components are linked by their geometric interfaces, and groups of components can be accurately identified and located in the DMU using their function and geo-

metric structure, e.g., adjusted bolted junctions with (screw+nut) (see Figure 6.8). Now, the geometric transformations needed to adapt the DMU to FEA objectives are strengthened because screws, nuts, locking nuts can be robustly identified, groups of tightened components are also available through the load cycles attached to cap screws (see Section 3.3).

Two possible accesses are proposed to define a function-based template T related to an assembly function:

- A component C through a user-defined selection: from it and its functional designation, a data structure gives access to the functions it contributes to. After selecting C, the user selects the function of interest among the available functions attached to C in $\mathcal{T}_{FN}$ and compatible with T. Other components are recovered through the selected function this component participates to;
- The function itself in $\mathcal{T}_{FN}$ that can lead to the set of components needed to define this function and all the instances of this function existing in the targeted assembly.

These accesses can be subjected to constraints that can help identifying the proper set of instances. Constraints aim at filtering out instances when a template T is defined from a function to reduce a set of instances down to the users needs, e.g., assembly function with bolts '*constrained with*' 2 tightening plates $component_i$ and $component_j$. Constraints aim at extending a set of instances when a template is defined from a component, i.e., a single function instance recovered, and needs to be extended, e.g., assembly function with bolts '*constrained with*' same tightened components and screw head functional interface of type '*planar support*' or '*conical fit*'.

6.3.3.1 Function-based template and compatibility conditions of transformations

The previous section has sketched how component functions can be used to identify sets of components in an assembly. Indeed, this identification is based on classes appearing in $\mathcal{T}_{FN}$. Here, the purpose is to define more precisely how the template can be related to $\mathcal{T}_{FN}$ and what constraints are set on shape transformations to preserve the geometric consistency of the components and their assembly. Shape transformations are application-dependent and the present context is structural mechanics and FEA to define a range of possible transformations.

The simplest relationship between a template T and $\mathcal{T}_{FN}$ is to relate T to a leaf of $\mathcal{T}_{FN}$. In this case, T covers instances defining sets of components that contain a variable number of components. T is also dimension independent since it covers any size of component, i.e., it is a parameterized entity. Shape transformations on T are designated as $S_{\mathcal{T}}$ and the template devoted to an application becomes $S_{\mathcal{T}}(T)$. Now, reducing the

scope to disassemblable assembly functions and more specifically bolted junctions, one leaf of $\mathcal{T}_{FN}$ can be used to define more precisely T and $S_{\mathcal{T}}(T)$. Conforming to Figure 6.8, let us restrict first to the leaf '*screw+nut*' of $\mathcal{T}_{FN}$. Then, T contains the following functional interfaces: one threaded link, two or more planar supports (one between nut and plate and at least one between two plates), either one planar support or one conical fit between the screw head and a plate, as many cylindrical loose fits as plates between the screw and plates because the class of junctions is of type adjusted. The shape transformations $S_{\mathcal{T}}(T)$ of T set up to process bolted junctions can be summarized as (see Figure 6.9):

- $S_{\mathcal{T}1}$: merging screw and nut (see Section 6.3.3.1);
- $S_{\mathcal{T}2}$: localization of friction effects with a sub-domain around a screw (see Section 6.3.3.1);
- $S_{\mathcal{T}3}$: removal of the locking nut if it exists (see Section 6.3.3.2);
- $S_{\mathcal{T}4}$: screw head transformation for mesh generation purposes (see Section 6.3.3.3);
- $S_{\mathcal{T}5}$: cylindrical loose fit around the screw shaft to support the contact condition with tightened plates (see Section 6.3.4).

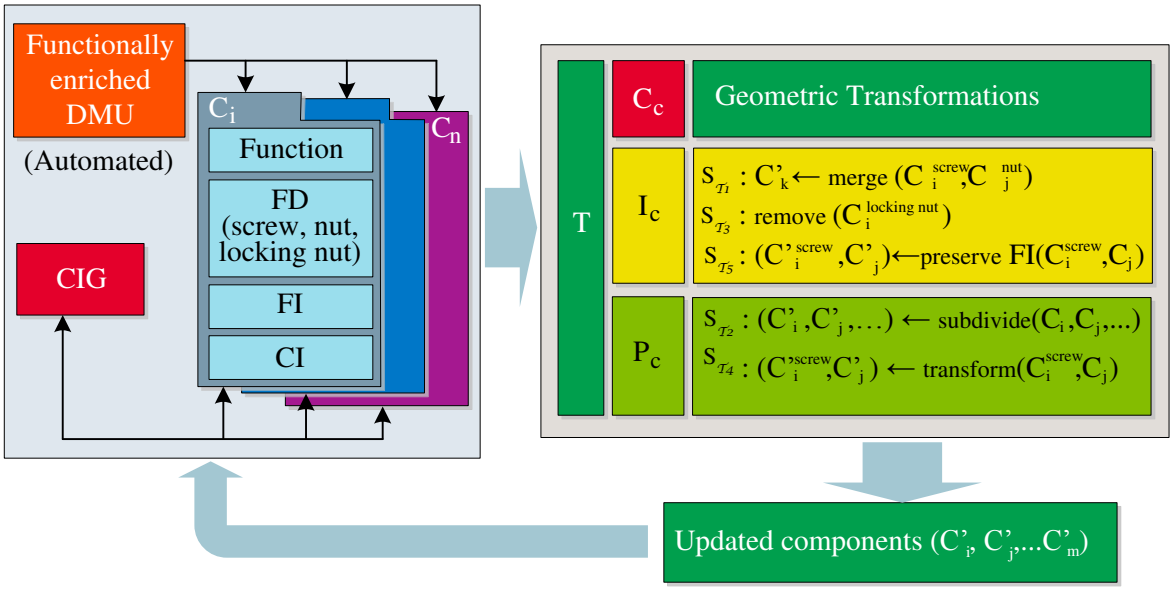
Each of these transformations are detailed throughout the following sections.

Now, the purpose is to define $S_{\mathcal{T}}$ so that $S_{\mathcal{T}}(T)$ exists and preserves the consistency of the components and the assembly. This defines compatibility conditions, C_C, between T and $S_{\mathcal{T}}$ that are conceptually close to attachment constraints of form features on an object [vdBvdMB03] (see Figure 6.9). C_C applied to $S_{\mathcal{T}}$ are introduced briefly here. Given the set of components contained in T, this set can be subdivided into two disjoint subsets as follows:

- I_C is the set of components such that each of its components has all its functional interfaces in T, e.g., the screw belongs to I_C. Consequently, components belonging to I_C are entirely in T (see the green rectangle in Figure 6.9);
- P_C is the set of components such that each of its components has some of its functional interfaces in T, e.g., a plate belongs to P_C. Components belonging to P_C are partially in T (see the red rectangle in Figure 6.9).

I_C can be used to define a 3D sub-domain of T, T_I defined as the union of all components belonging to I_C. Now, if a transformation $S_{\mathcal{T}}$ takes place in I_C and geometrically lies inside T_I , $S_{\mathcal{T}}(T)$ is valid because it cannot create interferences with other components of the assembly, i.e., C_C are satisfied.

Let us consider some of these transformations to illustrate some C_C. As an objective of FEA, the purpose of the assembly model is to analyze the stress distribution between



C_i : component i
CI: conventional interface
CIG: conventional interface graph
CC: compatibility conditions between T and $S_{\mathcal{T}}$;
FD: functional designation of a component;
FI: functional interface
I_C: set of components such that each of its components has all its FI in T
P_C: set of components such that each of its components has some of its FI in T
$S_{\mathcal{T}}$: shape transformation incorporated in a template, hence $S_{\mathcal{T}}(T)$;
T: function-based template performing shape transformations;

Figure 6.9: Principle of the template-based shape transformations. The superscript of a component C, if it exists, identifies its functional designation.

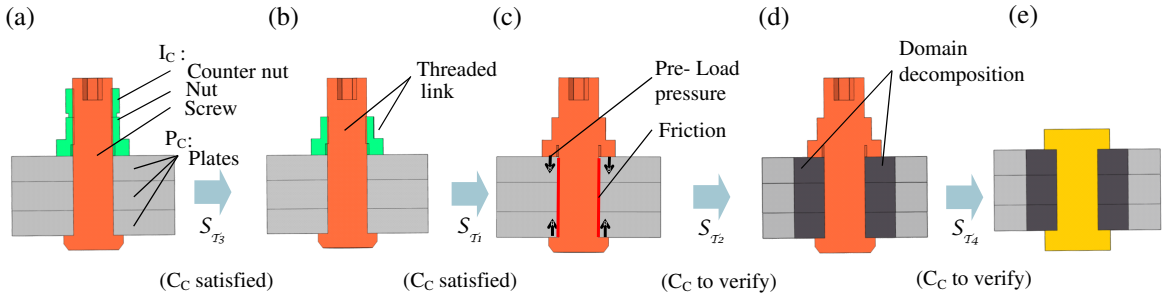


Figure 6.10: Compatibility conditions (C_C) of shape transformations $S_{\mathcal{T}}$ applied to T. (a) through (e) are different configurations of C_C addressed in Section 6.3.3.

plates and their interactions with bolts. To this end, the stress distribution around the threaded link between a screw and a nut is not relevant. Therefore, one shape transformation, $S_{\mathcal{T}1}$ is the removal of the threaded link to merge the screw and the nut (see Figure 6.10c). $S_{\mathcal{T}1}$ is always compatible since the screw and the nut belong to I_C, hence the C_C are always valid.

Now, let us consider another transformation, $S_{\mathcal{T}2}$, that specifies the localization of friction effects between plates around the screw shaft and the representation of the stress distribution nearby the screw shaft. This is modeled with a circular area centered on the screw axis and a cylindrical sub-domain around the screw shaft (see Figure 6.10d). Indeed, $S_{\mathcal{T}2}$, is a domain decomposition [CBG08, Cha12] taking place in the plates belonging to T. Because the plates belong to P_C, C_C are not trivial. However, $S_{\mathcal{T}2}$, takes place inside the plates so they cannot interfere with other components, rather they can interfere with the boundary of the plates or they can interfere between them when several screws are close to each other on the same plate (see Figure 6.11). In this case, C_C can be simply expressed, in a first place, as a non interference constraint.

Other shape transformations are listed when describing one example template in Section 6.3.4.

6.3.3.2 Shape transformations and function dependency

The previous section has connected T to $\mathcal{T}_{FN}$ in the simplest way possible, i.e., using a leaf that characterizes a single function. The purpose of this section is to analyze into which extent T can connect to classes of $\mathcal{T}_{FN}$ that perform several functions in the assembly.

In a first place, let us review shortly some concepts of functional analysis [Ull09]. There, it is often referred to several categories of functions that are related to a design process, i.e., external, internal, auxiliary, ... However, this does not convey consistency conditions among these functions, especially from a geometric point of view. Here, the current content of $\mathcal{T}_{FN}$ refers to internal functions, i.e., functions strictly performed by components of the assembly. The '*screw+nut*' function, as part of bolted junctions, is one of them. Bolted junctions can contain other functions. Let us consider the locking function, i.e., the bolt is locked to avoid any loss of tension in the screw when the components are subjected to vibrations. The locking process can take place either on the screw or on the nut. For the purpose of the analysis, we consider here a locking process on the nut, using a locking nut (see Figure 6.12a). In functional analysis, this function is designated as auxiliary function but this concept does not characterize geometric properties of these functions.

From a geometric point of view, it can be observed that functional interfaces of the screw, nut and locking nut are located in 3D such that the functional interfaces (planar support) between the nut and locking nut cannot exist if the nut does not tighten the plates. Consequently, the locking function cannot exist if the tightening

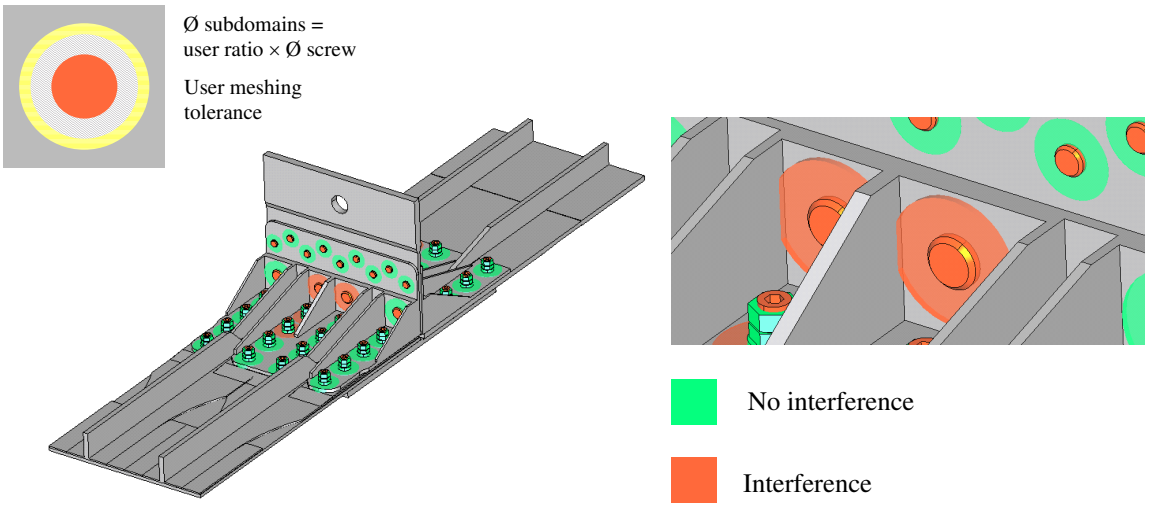


Figure 6.11: Checking the compatibility of $S_{\mathcal{T}}(T)$ with respect to the surrounding geometry of T.

function does not exist. Rather than using the designation of auxiliary function, which is geometrically imprecise, it is referred to *dependent function*.

The concept of dependent functions is inserted in $\mathcal{T}_{FN}$ at different levels of $\mathcal{T}_{FN}$ to attach the corresponding functions when they exist (see Figure 6.8). Based on the concept of dependent function, it is possible to extend the connection rule between T and $\mathcal{T}_{FN}$. Rather than connections at the leaf level, higher level classes can be connected to T if the dependent functions are taken into account in the C_C of shape transformations $S_{\mathcal{T}}$ so that $S_{\mathcal{T}}(T)$ exists and preserves the consistency of the assembly. As an illustration, let us consider T connected to '*Bolted adjusted*' (see Figure 6.8). Now, $S_{\mathcal{T}}$ can cover the class of bolted junctions with locking nut. Let $S_{\mathcal{T}3}$, be the transformation that removes the locking nut of a bolted junction, which meets also the FEA objectives mentioned earlier. Because $S_{\mathcal{T}3}$, applies to a dependent function of '*screw+nut*', the C_C are always satisfied and the resulting model has a consistent layout of functional interfaces, i.e., the removal of the locking nut cannot create new interfaces in the assembly (see Figure 6.9 and 6.10b). Consequently, T can be effectively connected to '*Bolted adjusted*', which is a generalization of T.

6.3.3.3 Template generation

T is generated on the basis of the components involved in its associated function in $\mathcal{T}_{FN}$. T incorporates the objectives of the FEA to specify $S_{\mathcal{T}}$. Here, $S_{\mathcal{T}}$ covers all the transformations described previously, i.e., $S_{\mathcal{T}1}$, $S_{\mathcal{T}2}$, $S_{\mathcal{T}3}$. Figure 6.10 and 6.11 illustrates the key elements of these shape transformations.

Other shape transformations, $S_{\mathcal{T}4}$, can be defined to cover screw head transformations and extend the range of screws to flat head ones. However, this may involve

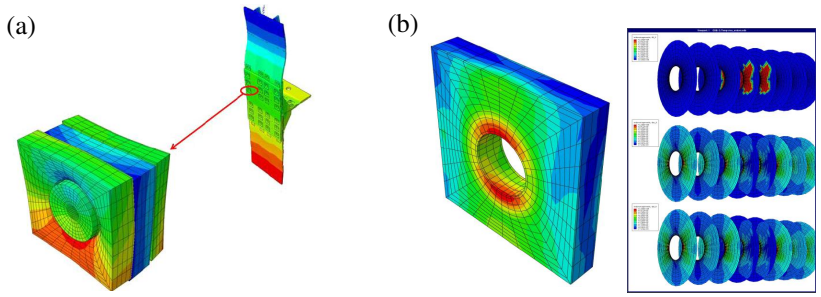


Figure 6.12: (a) Multi-scale simulation with domain decomposition around bolted junctions, (b) Load transfers at critical holes (courtesy of ROMMA project [ROM14]).

geometric transformations where the volume of a screw head gets larger. In this case, $S_{\mathcal{T}4}$ takes place in P_C and the compatibility conditions are not intrinsic to T (see Figure 6.9). Consequently, it is mandatory to perform an interface/interference checking with the other components of the assembly to make sure that the transformation is valid (see Figure 6.10).

Then, the set of shape transformations structures the dialog with the user to allow him, resp. her, to select some of these transformations. However, the user settings are applied to instances whenever possible, i.e., when the instance belongs to a class where the shape transformations are applicable.

6.3.4 Example of template-based operator of bolted junctions transformation

In an aeronautical company, simulation engineers perform specific FEAs on assembly sub-structures such as the aircraft junction between wings and fuselage. Based on pre-existing physical testing performed by ROMMA project [ROM14] partners, this structure can be subjected to tensile and compressive forces to analyze:

- The distribution of the load transfer among the bolted junctions;
- The admissible extreme loads throughout this structure.

From the physical testing and preliminary numerical models, the following simulation objectives have been set up that initiate the requirements for the proposed template-based transformations. To adapt the FE model to these simulation objectives while representing the physical behavior of the structure, an efficient domain decomposition approach [CBG08, Cha12] uses a coarse mesh far enough from the bolted junctions and a specific sub-domain around each bolted junction with friction and preload phenomena (see Figure 6.12a, b). The objective is not to generate a detailed stress distribution everywhere in this assembly but to observe the load distribution areas among bolts using the mechanical models set in the sub-domains.

The objective of this section is to validate the methodology of Section 6.2 through the template-based approach. The proposed demonstrator transforms automatically the bolts into simplified sub-domains ready for meshing with friction areas definition while preserving the consistency of the assembly. Consequently, there is no area of weak interest. All the above transformations are aggregated into a parameterized template whose input is the functional designation of components to locate the cap screws in the assembly. Then, the template adapts to the screw dimensions, the number of plates tightened, ..., to apply the operators covering tasks 2 through 6. The template features are aligned with the needs for setting up a simulation model able to exhibit some of the physical phenomena observed during testing and expressed in the above simulations results.

Operator description

Having enriched the assembly with functional information, the template interface lets the engineer select a node of $\mathcal{T}_{FN}$ that is compatible with T. In this example, the function to select is: '*assembly with Bolted junction*' (see Figure 6.14). Now, several $S_{\mathcal{T}}$ are either pre-set or user-accessible.

Figures 6.6a and 6.13 illustrates the task 2 of the methodology. An hypothesis focuses on the interfaces between screw shafts and plate holes: the clearances there are regarded as small enough in the DMU to be reduced to a fitted configuration where shafts and holes are set to the same nominal diameter to produce a conform mesh with contact condition at these interfaces. To precisely monitor the stress distribution around bolts and friction between plates, $S_{\mathcal{T}2}$ is user-selected. It a simplified model of the Rotschers cone [Bic95] that enables generating a simple mesh pattern around bolts. In this task also, hypothesizing that locking nuts, nuts, and screws can be reduced to a single medium leads to the removal of assembly interfaces between them. $S_{\mathcal{T}3}$ is user-accessible and set here to remove the dependent function '*locking with locking nut*'.

Then, there is no idealization taking place, hence no action in tasks 3 and 5. Task 4 connects to a hypothesis addressing the interfaces resulting from task 2, i.e., the interfaces between plates need not to model contact and friction over the whole interface. Friction effects can be reduced to a circular area around each screw, which produces a subdivision of these interfaces. $S_{\mathcal{T}5}$ is pre-set in T to preserve the cylindrical loose fit between screw and plates to set up contact friction BCs without inter-penetration over these functional interfaces. $S_{\mathcal{T}1}$ is also pre-set as well as $S_{\mathcal{T}4}$.

Finally, task 6 concentrates on skin and topological transformations. These are achieved with locking nut, nut, and screw shape transformations. The latter is performed on the functional interface (planar support) between the screw head/nut and the plates to obtain a meshing process independent of nut and screw head shapes. Now, T can cover any bolted junction to merge screw, nut and locking-nut into a single do-

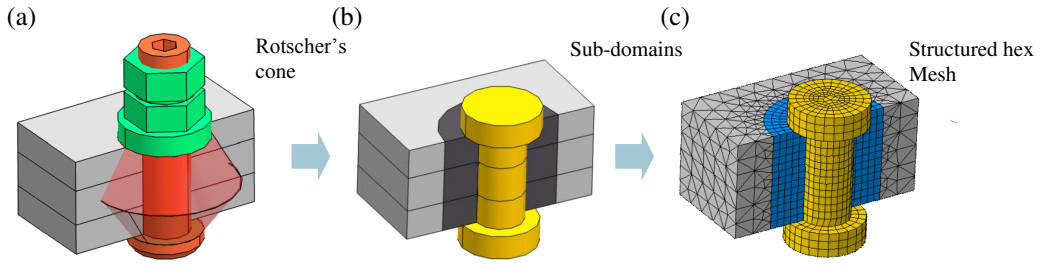


Figure 6.13: Template based transformation $S_{\mathcal{T}}(T)$ of a bolted junction into simple mesh model with friction and contact areas definition around screw and nut. (a) the bolted junction in the DMU, (b) the bolted junction after simplification to define the simulation model, (c) the desired FE mesh of the bolted junction.

main, reduce the screw and nut shapes to a simple shape of revolution while preserving the consistency of its interfaces.

Based on T, $S_{\mathcal{T}}(T)$ is fairly generic and parameterized to intelligently select and transform bolts, i.e., it is independent of the number and thicknesses of plates, of the screw diameter, the length and head type (cylindrical (see Figure 6.13a) versus flat ones) in addition to the location of each bolt.

Here, $S_{\mathcal{T}}(T)$ contains $S_{\mathcal{T}2}$, a generation of sub-domains taking into account the physical effects of the Rotschers cone. This geometric transformation could interact with plate boundaries to change the shape of these sub-domains and influence the mesh generation process. Presently, templates are standalone entities and are not taking into account these effects left for future developments. At present, the engineer can adjust the sub-domain to avoid these interactions (see Figure 6.14a).

Implementation and results

The developed prototype is based on OpenCascade [CAS14] and Python scripting language. The DMU is imported as STEP assembly models, the geometric interfaces between components are represented as independent trimmed CAD faces with identifiers of the initial face pairs of the functional interfaces. The assembly functional description is imported as a text file from the specific application performing the functional enrichment described in [SLF*13] and linked to the assembly model by component identifiers.

Figure 6.14a shows the user interface of the prototype. When selecting the '*assembly with Bolted Junction*' function, the user has a direct access to the list of bolted junctions in the assembly. To allow the user filtering his selection, DMU parameters are extracted from the functional designation of components, e.g., the screw and nut types, the number of tightened components, or from geometry processing based on functional interfaces, e.g., screw diameter. Using these parameters, the user is able

to select bolted junctions within a diameter range, e.g., between 10 and 16 mm (see Figure 6.14b) or bolted junctions with screw and locking nut (see Figure 6.14c), etc. The user can monitor the Rotschers cone dimension with a FEA parameter called '*sub domain ratio*' that represents the ratio between the screw nominal diameter and the sub-domain diameter (see Figure 6.14d and e). Then, the user-defined '*meshing tolerance*' is used during the verification phase to check the compatibility conditions, C_C, between instances and their surrounding geometry (see Figure 6.9 and 6.11).

Figure 6.15 shows two results of the template-based transformations on aircraft structures:

Aircraft structure 1: A junction between the wing and the fuselage. The assembly contains 45 bolted junctions with 3 different diameters and 2 different screw heads;

Aircraft structure 2: An engine pylon. The assembly contains over 250 bolted junctions with identical screws and nuts.

The final CAD assembly (see Figure 6.14b) with simplified bolted junctions has been exported to a CAE software, i.e., Abaqus [FEA14]. STEP files [ISO94, ISO03] transfer the geometric model and associated xml files describes the interfaces between components to trigger meshing strategies with friction area definitions. Appendix D illustrates the STEP data structure used to transfer the Aircraft structure 1.

Comparing with the process pipeline used with existing industrial software (see Section 1.4.3), the improvements are as follows. The model preparation from CAD software to an Abaqus simulation model takes 5 days of interactive work for 'Aircraft structure 1' mentioned above (see Figure 1.13b). Using the pipeline performing the functional enrichment of the DMU and the proposed template-based shape transformations to directly produce the meshable model in Abaqus and perform the mesh in Abaqus, the overall time is reduced to one hour. The adequacy of this model conforms to the preliminary numerical models set up in ROMMA project [ROM14] and extending this conformity to testing results is ongoing since the template enables easy adjustments of the mesh model.

Regarding the 'Aircraft structure 2', there is no reference evaluation of its model preparation time from CAD to mesh generation because it is considered as too complex to fit into the current industrial PDP. However, it is possible to estimate the time reduction since the interactive time can be linearly scaled according with the number of bolted junction. This ends up with 25 days of interactive work compared to 1.5 hour with the proposed approach where the time is mostly devoted to the mesh generation phase rather than the template-based transformations.

Though the automation is now very high, the template-based approach still leaves the engineer with meaningful parameters enabling him/her to adapt the shape transformations to subsets of bolted junctions when it is part of FE requirements.

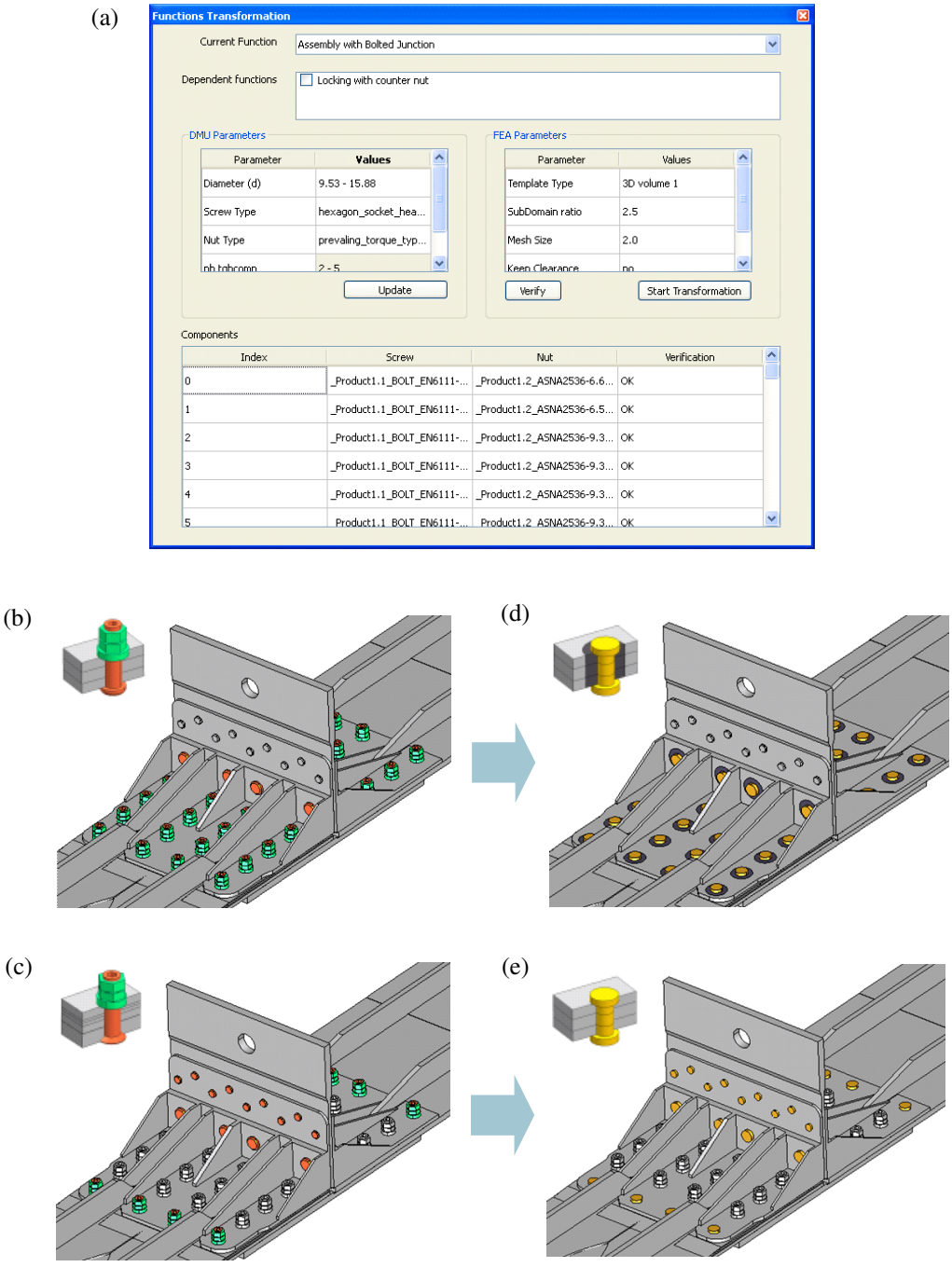


Figure 6.14: (a) User interface of a template to transform '*assembly Bolted Junctions*'. Results obtained when filtering bolts based on diameters (b) or screw type (c). Results of the template-based transformations with (d) or without (e) sub-domains around bolts.

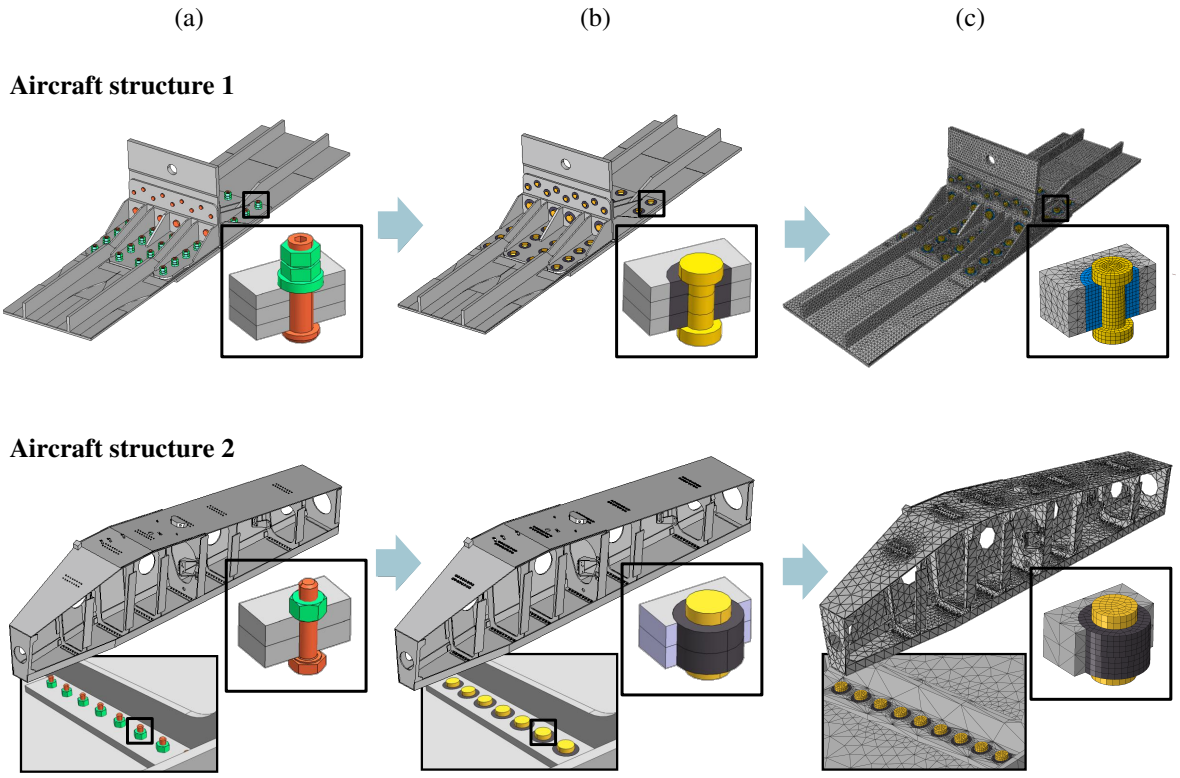


Figure 6.15: Results of template-based transformations on CAD assembly models: (a) CAD models with functional designations and geometric interfaces, (b) models (a) after applying $S_{\mathcal{T}}(T)$ on bolts, (c) mesh assembly models obtained from (a) with friction area definition.

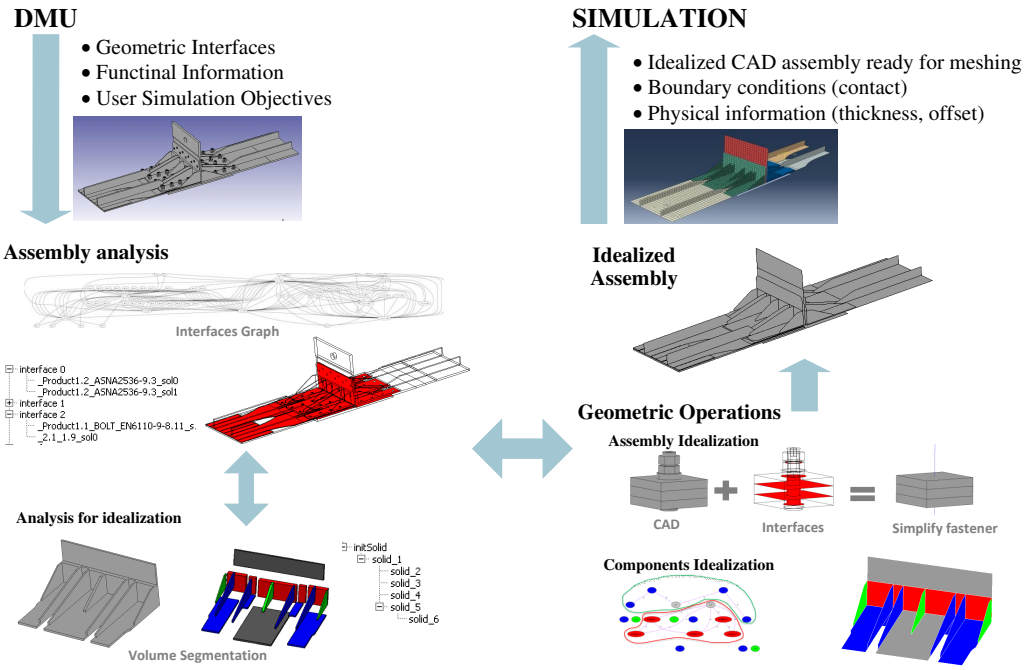


Figure 6.16: Illustration of the idealization process of a CAD assembly model. All components are fully idealized and bolted junctions are represented with FE fasteners. Solid plates and stiffeners are idealized as surfaces.

6.4 Full and robust idealization of an enriched assembly

The methodology of Section 6.2.2 has also been applied to create an idealized plate model of the 'Root joint' use-case presented in Figure 1.6. The simulation objectives are set on the global analysis of the stress field in the structure and the analysis of the maximal loads transferred through the bolted junctions. Consequently (see Section 1.4.3), the generated FEM contains idealized components with shell FE and each junction can be simplified with a fastener model (see Figure 6.17). Figure 6.16 illustrates the different data and processes used to transform the initial DMU model into the final FE mesh model. Once the CAD data associated with functional interfaces have been imported, all bolted connections are transformed into simplified fastener models (see Section 6.4.1) in a first step. Then, a second step segments and idealizes all components in accordance with the method described in Chapter 5 (see Section 6.4.2).

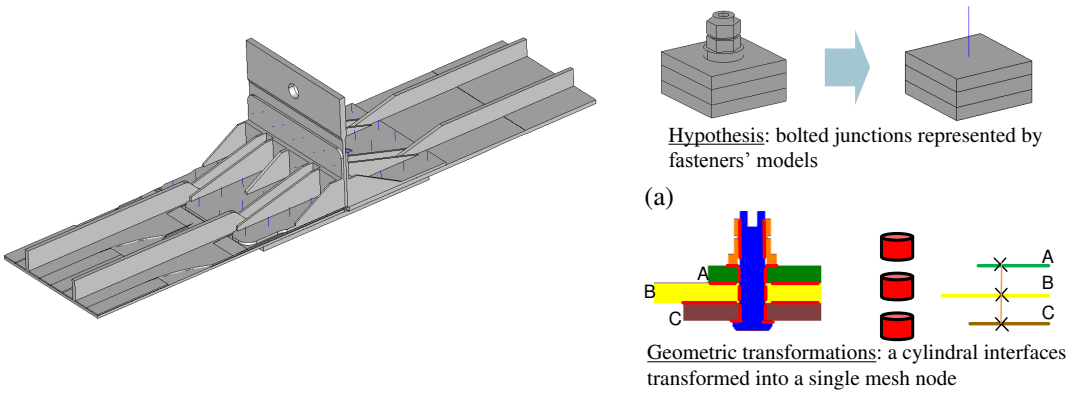


Figure 6.17: Illustration of Task 2: Transformation of bolted junction interfaces into mesh nodes.

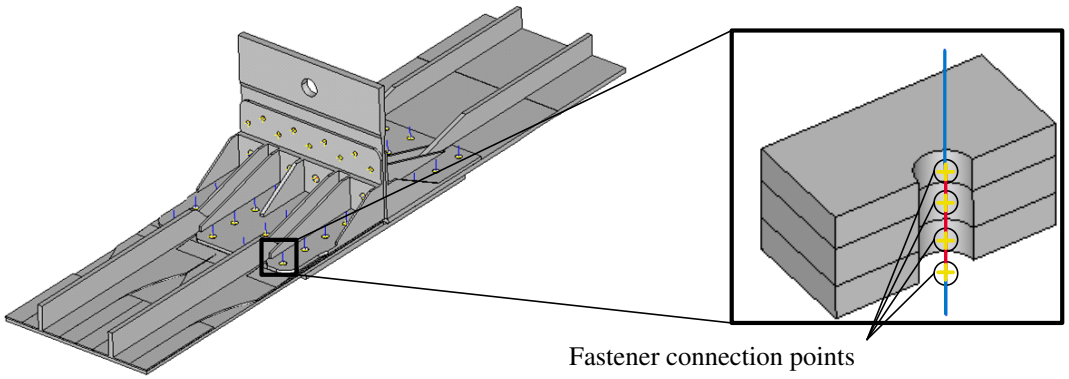


Figure 6.18: Results of the template-based transformation of bolted junctions. The blue segment defines the screw axis. Red segments are projections of interfaces between the plates and the screw. Yellow points are the idealizations of these interfaces to connect the screw to the plates.

6.4.1 Extension of the template approach to idealized fastener generation

Given the above simulation objectives, the first step of transformations is related to the transfer of plate loads throughout the assembly and FE beam elements are sufficient to model the bolts behavior. This hypothesis implies, in task 2 of the methodology, a transformation of cylindrical interfaces between bolts and plate holes into single mesh nodes (see Figure 6.17a) linked to beam elements that represent the idealized fasteners. A specific template-based operator has been developed to automate the transformation of bolted junctions into idealized fasteners. As in Section 6.3, the bolted junctions are identified among all 3D components through the function they are involved in: 'adjusted bolted junctions with screw+nut'. Then, the template applies a set of shape transformations $S_{\mathcal{T}}$ to generate the beam elements with their associated connection points. These shape transformations are described as follows:

- $S_{\mathcal{T}1}$: merging screw and nut (identical to Section 6.3.3.1);
- $S_{\mathcal{T}2}$: removal of the locking nut if it exists (identical to Section 6.3.3.2);
- $S_{\mathcal{T}3}$: screw transformation into beam elements (see Figure 6.18). FE beam elements are represented by line segments. The axis of the screw is used as location for the line segments;
- $S_{\mathcal{T}4}$: transfer of interfaces between plates and screw as points on the line segments. The blue part in Figure 6.18 represents the whole screw, the red parts are the projections of the interfaces between the plates and the screw while yellow points represent the idealization of these interfaces and define the connections between each plate and the screw;
- $S_{\mathcal{T}5}$: reduce junction holes in plates to single points. The fastener model used to represent bolted junctions does not represent holes in the final idealized model (see Figure 6.19).

Different idealizations, i.e., the set of $S_{\mathcal{T}}$, can be generated to match different simulation objectives. For instance, one $S_{\mathcal{T}}$ can focus on the screw stiffness in addition to the current simulation objectives. To this end, a new objective can be set up to compare the mechanical behavior when screws are modeled as beams and when they are perfectly rigid. Consequently, the screws must now be represented as rigid bodies. This means that the blue part (see Figure 6.18) representing the FE beam element is no longer needed. Then, the yellow points (see Figure 6.18) can be used directly to generate the mesh connection points in the final medial surfaces of the plate components. These points can be used to set kinematic constraints.

Indeed, the list of previous geometric operations describes a new category of shape transformation, $S_{\mathcal{T}i}$ that would be needed to meet this new simulation objective. Be-

cause these transformations are close the templates described, this shows how the principle of the template-based transformations can be extended to be adapted to new simulation objectives using additional elementary transformations. Here, $S_{\mathcal{T}3}$ would be replaced by $S_{\mathcal{T}i}$ that would produce the key points needed to express the kinematic constraints.

Now that the bolted junctions have been simplified into FE fasteners, the next section illustrates the idealization of the whole assembly.

6.4.2 Presentation of a prototype dedicated to the generation of idealized assemblies

In order to generalize the idealization approach presented in Chapter 5, a prototype has been developed to process not only components but also whole assemblies. Likewise the template demonstrator, the prototype is based on OpenCascade [CAS14] and Python scripting language. The CAD assembly as well as the geometric interfaces are imported as STEP models.

Figure 6.19 illustrates the user interface of the prototype. Here, the 3D viewer shows the result of task 2 where the bolted junctions of the CAD assembly have been transformed into simple fasteners using the template-based approach. The interface graph in the graph tag shows all the neighboring relationships between assembly components (including the fasteners).

Figure 6.20 illustrates task 3 where the assembly components are segmented into sub-domains according to shape primitives organized into a construction graph. The set of solid primitives in red is extracted using the algorithm described in Section 4.5.2. The primitives are then removed from the initial solid to obtain a simpler component shape. Once the construction graph is generated, the user selects a construction process which creates a component segmentation into volume sub-domains. Then, each sub-domain is idealized wherever the primitive extent versus its thickness satisfies the idealization criterion. The interfaces resulting from this idealization can be associated with new transformations of assembly interfaces, e.g., a group of parallel idealized surfaces linked to the same assembly interface can be aligned and connected. The analysis of interactions between independently idealized sub-domains can guide geometric transformations such as sub-domain offsets and connections. These transformations are part of task 4. Figure 6.21 illustrates an intermediate result of the idealization process of a component (task 5). The graph of primitives' interfaces has been analyzed in task 4 to identify and align groups of parallel medial surfaces. For example, the medial surface highlighted in brown is offset by 2.9 mm from its original position. The medial surfaces are then connected with the operator described in Section 5.5.3. The result is a fully idealized representation of the component.

Finally, other idealized components are incorporated in the idealized assembly

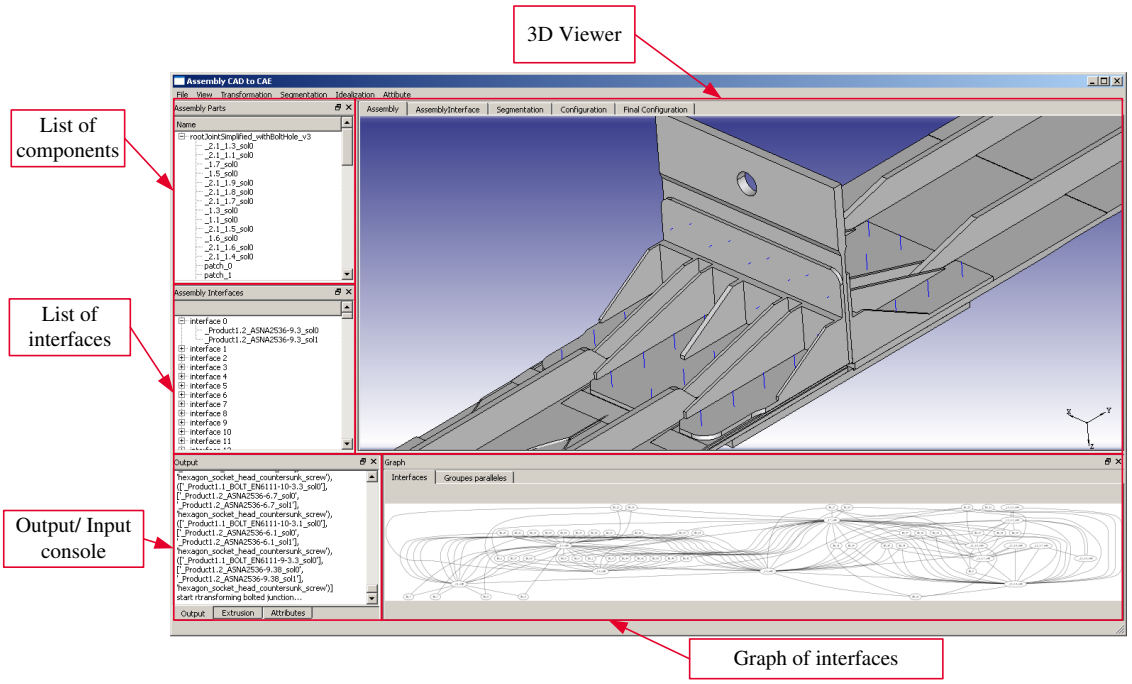


Figure 6.19: User interface of the prototype for assembly idealization. The 3D viewer shows the assembly after the transformation of bolted junctions into FE fasteners.

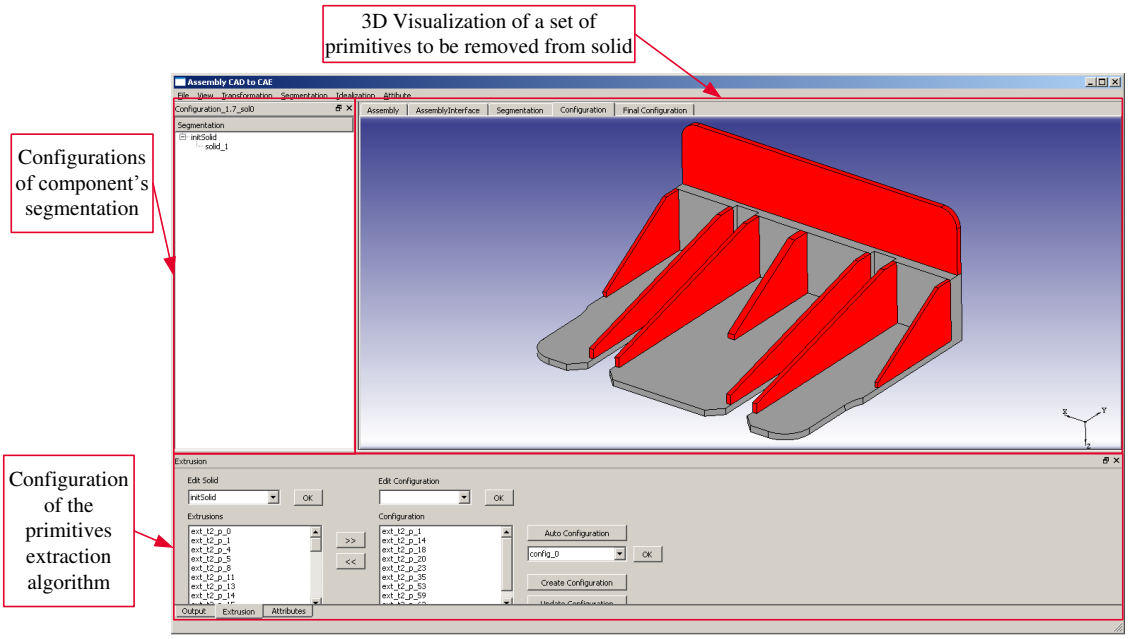


Figure 6.20: Illustration of a component segmentation which extract extruded volumes to be idealized in task 3. The primitives to be removed from the initial solid are highlighted in red.

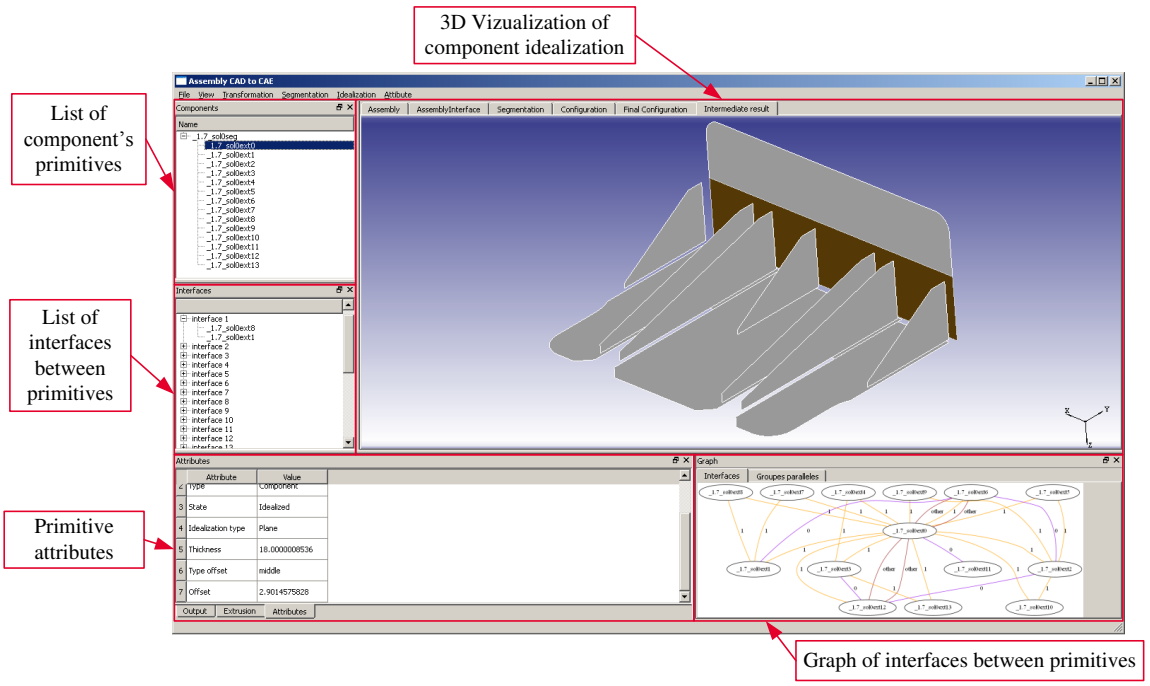


Figure 6.21: Illustration of task 4: Identification and transformation of groups of idealized surfaces connected to the same assembly interfaces.

model using complementary sub-domain transformations applied to each of them as illustrated in Figure 6.22.

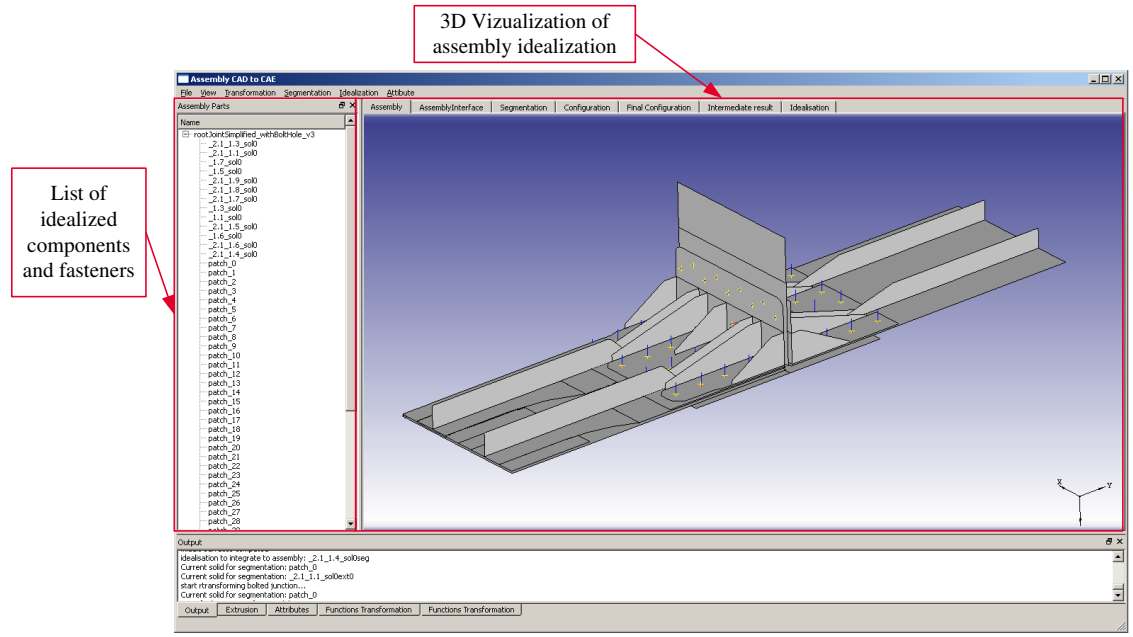


Figure 6.22: Final result of the idealized assembly model ready to be meshed in CAE software.

Again, functional information about components and successive decomposition into

sub-domains as well as idealization processes reduce the preparation process to minutes: up to approximately ten minutes to process all the components including all the user interactions required to load each component, select the appropriate template or process the components subjected to the segmentation process and the morphological analysis. This is a significant time reduction compared to the days required when performing interactively the same transformations using tedious interactions with low level operators existing in current CAE software.

Yet, the constraints related to mesh generation through mesh size constraints, have not been taken into account in the current analysis of preparation processes. These constraints have to be addressed in future research work. It has also to be noted that the process flow of Figure 6.5 turns into a sequential flow in all the simulation frameworks illustrated.

6.5 Conclusion

In this chapter, dependencies between categories of shape transformations have been organized to structure the assembly simulation model preparation process in terms of methodology and scope of shape transformation operators. The proposed methodology empowers the use of DMUs enriched with geometric interfaces and functional information to automate CAD assembly pre-processing and to generate a 'FE-friendly' equivalence of this assembly.

The template-based method has shown that shape transformations highly benefit from functional information. Using functional information strengthens the transformation of complex models like assemblies where many components interact with each other. The template can be instantiated over the whole assembly to quickly transform repetitive configurations such as bolted junctions that are highly time-consuming when processed purely interactively.

The idealization method introduces a robust geometric operator of assembly idealization. This operator takes advantage of the assembly decomposition into sub-domains and their associated geometric interfaces as produced in Chapter 5. This structure has been used successfully to idealize sub-domains and address some general mesh generation constraints to ensure obtaining high quality meshes.

Finally, a demonstrator has been implemented to prove the validity of the proposed methodology. This prototype has been applied to an industrial use-case proposed in ROMMA project [ROM14] to create a simplified solid model and an idealized, surface-based model, using the operators currently developed. This use-case has demonstrated the benefits of the proposed methodology to:

1. Efficiently process real 3D assemblies extracted from large DMU;

2. Enable the implementation of a robust approach to monitor automated shape transformations.

Thanks to this methodology, the preparation time can be drastically shortened compared to purely interactive processes as commonly practiced by today's engineers.

Conclusion and perspectives

Assemblies, as sets of components, bring a new complexity level for CAD-FEA data processing of mechanical structures. Here, new principles and associated operators have been developed to automate the adaptation of CAD assembly models. The objective targeted robust transformations to process the large amount of repetitive geometric configurations of complex assemblies and to reduce the time spent by engineers to prepare these assemblies.

Summary of conclusions

Now, each of the contributions stated in the previous chapters can be synthesized and summarized.

In-depth analysis of FE pre-processing rules

The first contribution stands in the analysis of the current pre-processing of CAD models derived from DMUs to produce FE models. Due to the lack of assembly-related information in a DMU, very tedious tasks are required to process the large amount of components as well as their connections. Preparing each component is already a tedious task, especially when idealizations are necessary, that increases significantly with the number of components and their interfaces. Additionally, these interfaces form new entities to be processed. It has been observed that repetitive configurations and their processing are also an issue of assembly preparation, justifying the need to automate the preparation of large assembly models. This first analysis has concluded that the adaption of an assembly to FEA requirements and geometric transformations derive from simulation objectives and component functions are needed as well to geometrically transform groups of components. Also, it has been shown that functional information can be an efficient enrichment of a DMU to identify and process repetitive configurations.

Challenging assembly models preparation

Studying the current CAD-FEA methods and tools related to data integration

reveals that operators currently available focus on the transformation of standalone components. One main contribution of this thesis is the proposal of an approach to assembly model preparation. Rather than reducing the preparation process to a sequence of separately prepared parts, the entire assembly has been considered when specifying shape transformations to reach simulation objectives, taking into account the kinematics and physics associated with assembly interfaces. The proposed approach to assembly pre-processing uses, as input model, a DMU enriched at an assembly level with interface geometry between components, additional functional properties of these components, and, at the component level, a structured volume segmentation using a graph structure.

Geometrically enriched components

A new approach has been proposed to decompose a B-Rep solid into volume sub-domains. This approach robustly enriches a CAD component using a construction graph and provides a volume decomposition of each component shape. This construction graph is generated by iteratively identifying and removing a set of extrusion primitives from the current B-Rep shape. It has been shown that, compared to any initial construction process performed by a designer, the extracted graph is unique for a given object and is intrinsic to its shape because it overcomes modeling, surface decomposition, and topological constraints. In addition, it provides non trivial construction trees, i.e., variants of extrusion directions producing the same primitive are not represented and variants in primitives ordering are grouped into a single node. This generates a compact representation of a large set of shape construction processes. Moreover, the proposed approach, while enriching a standalone component shape, can be extended to assembly structures after they have been enriched with component interfaces. Each component construction graph can be nested into the component-interface assembly structure, thus forming a robust data structure for CAD-FEA transformation processes. Finally, a graph of generative processes of a B-Rep component is a promising basis to gain a better insight about a shape structure. The criteria used to generate this graph bring meaningful and simple primitives which can be subsequently used to support the idealization process of component shapes.

Formalizing a shape idealization process through a morphological analysis

It has been shown that generating a fully idealized model cannot be reduced to a pure application of a dimensional reduction operator such that this model is a mechanical equivalence of the initial component. The incorporation of idealization hypotheses requires the identification of candidate geometric areas associated with the connections between idealized sub-domains. The proposed idealization process benefits from the new enrichment of components with their shape structure. The segmentation of the component into meaningful primitives and interfaces between them has been used as a first step of a morphological analysis. This analysis evaluates each primitive with re-

spect to its dominant idealized shape. Then, using a taxonomy of geometric interfaces between idealized sub-domains, this analysis is propagated over the whole component and results in the decomposition of its shape into 'idealizable' areas of type 'plate/shell, beam and 'non-idealizable' areas. Overall, the morphological analysis is independent from any resolution method and is able to characterize geometric details in relation to local and to 'idealizable' regions of a component. Finally, an idealization operator has been developed which transforms the sub-domains into medial surfaces/lines and robustly connects them using the precise geometric definition of interfaces between primitives.

Laying down the basis of a methodology to assembly preparation

To address the current industrial needs about assembly pre-processing for structural simulation, the analysis of dependencies between geometric transformations, simulation objectives, and simplification hypotheses led to a first methodology increasing the level of automation of FE assembly model pre-processing. Using an enriched DMUs containing geometric interfaces between components and their primitives as well as functional information to end up with the generation of a 'FE-friendly' equivalence of this assembly, the methodology is in line with the industrial needs to develop a new generation of DMU: the Functional DMU. Finally, the development of a prototype platform has illustrated that the methodology fits well with the methods and tools proposed in this thesis. The template-based transformation, empowering the use of functional information, has illustrated how repetitive configurations, such as assembly junctions, can be automatically transformed. Then, the generation of the complete idealization of an aeronautical structure has demonstrated the ability of the proposed idealization approach to efficiently process CAD assemblies extracted from a large DMU.

As a final conclusion, compared to purely geometric operators currently available in CAD-FEA integration, this thesis has proposed an approach based on a shape analysis of a enriched DMU model that significantly shortens the time commonly spent by today's engineers and robustly performs repetitive idealization transformations of components and assemblies as well.

Research perspectives

From the proposed approach of DMU pre-processing for structural assembly simulation, future work can extend and build further on the methods and tools described in this thesis. The perspectives presented in this section refers to the generation of construction graph of B-Rep shapes and to the morphological analysis of DMU models.

Construction graph

Regarding the generation of construction graphs from B-Rep shapes, perspectives are listed as follows:

- **Extend the definition of primitive to include material removal as well as additional operations (revolution, sweep,. . .).** In a first step, to reduce the complexity of this research work, the choice has been made to concentrate the extraction of generative processes on extrusion primitives. Primitives are combined solely using a material addition operator. Clearly, future work will focus on incorporating material removal operations and revolutions to extend the range of objects that can be processed. Allowing new definitions of primitives may increase the amount of primitives. However, the construction graph can be even more compact. Indeed, groups of extrusion primitives can be replaced by a unique revolution, or a sweeping primitive in the construction graph. To reduce the complexity of the dimensional reduction of primitives, the presented idealization process favored primitives adding material instead of primitives removing material. Including primitives which removes material can be convenient for other applications, e.g., to simplify components' shapes for 3D simulation or to identify cavities in components for computational fluid simulations.

- **Extend the attachment condition of primitives.** Regarding the attachment of a primitive into an object, it has been shown all the benefits to avoid constraining the primitive identification process with their attachment conditions and to avoid looking at prioritizing primitives with geometric criteria such as: largest visible boundaries within the object. Identifying primitives without restriction on their 'visible' boundaries is a way to release this constraint. However, to validate the major concepts of the proposed approach, two restrictions have been set on the primitive definition. The extrusion distance had to be represented by a lateral edge and one of the primitive's base face had to be totally 'visible'. A future target stands in the generalization of the primitive definition to enlarge the number of valid primitives and hence, will produce a more generic algorithm;

- **Reduce the interaction between primitives.** Currently, the computation time is highly dependent on the number of extracted primitives which are compared with each others. To reduce the complexity of the algorithm, future work may integrate the identification of repetitions and symmetries [LFLT14]. It is not only the global symmetries or repetitions, e.g., reflective symmetries valid at each point of an object, which may directly reduce the extent of a shape being analyzed, more frequently partial symmetries and repetitions are more efficient to identify specific relationships between primitives. Partial symmetries and repetitions initiated by the location of identical primitives convey a strong meaning from a shape point of view. They can be used after the extraction of primitives to generate groups of symmetrical/repetitive primitives or even before that

stage to help identifying primitives, e.g., selecting a set of base faces sharing the same plane and the same orientation. Finally, symmetries and repetitions are very relevant to structure an idealized model to propagate these shape structure information across the mesh generation phase.

- **Further applications of construction graphs.** A construction graph structures the shape of a B-Rep object independently from any CAD modeler. Applied to hex-meshing, a shape intrinsic segmentation into extrusion primitives extracted from its construction graph can be highly beneficial. Indeed, it directly provides simple meshable volumes. Moreover, the complementary information about the connections between primitive interfaces can help to generate a complete component 3D mesh. Applied to 3D direct modeling CAD software, this intrinsic shape structure can be used to significantly extend this approach with larger shape modification as well as parametrization capabilities. Because primitives are geometrically independent of each other, the parametrization of a primitive can be directly related to the object shape, i.e., the influence of the shape modification of a primitive can be identified through the interface of this primitive with the object.

Morphological analysis of assemblies.

The morphological analysis method of Chapter 5 has been presented as a preliminary step for dimensional reduction operations. The perspectives related to this morphological analysis are as follows:

- **Extend the taxonomy of reference morphologies.** The determination of idealizable volume sub-domains in a component is based on a taxonomy of morphologies. Each morphology is associated with one medial edge of the MAT applied to the extrusion contour of a primitive. Clearly, this taxonomy is not complete. Only morphologies associated with straight medial edges with constant radius has been studied. To enable the processing of a larger range of component shapes, this taxonomy can be extended, in a first step, to curved edges, with or without radius variation, and, in a second step, to other types of primitives (revolution, sweeping, ...);

- **Extend the taxonomy of connections.** Regarding the propagation of the morphological analysis of primitives to the whole object, the current taxonomy of connections covers extrusion sub-domains to be idealized with planar medial surfaces only. The detailed study of these configurations has demonstrated all the robustness of the proposed approach. Here too, this taxonomy can be enlarged to process beams, shells, or thick domains. In addition, the taxonomy of connections is currently restricted to couples of sub-domains. In case of groups of connected sub-domains, a new level of morphology may emerge, e.g., a set of piled up thin extrusion primitives forming a beam. Analyzing and formalizing

this new taxonomy of connections between sub-domains will enlarge the shape configurations which can be processed;

- **Extend the approach to the morphological analysis of assemblies.** Although construction graph structures (primitives/interfaces) is compatible with assembly structures (components/interfaces), the morphological analysis has been applied on standalone components. When the input model is an assembly structure, the assembly interfaces brings a new level of information. The influence of assembly interfaces on the established taxonomies have to be studied to extend the morphology analysis to assembly. For example, on large assemblies models, a group of component can be viewed as a unique morphology. Propagating the morphological analysis of components to the whole assembly will give the user a complete access to multi-resolution morphological levels of a DMU.

Finally, we can mention the report from a 2013 ASME Panel on Geometric Interoperability for Advanced Manufacturing [SS14]. The panelists involved had considerable experience in the use of component geometry throughout various design and manufacturing softwares. They stated that current CAD systems have hit hard limit with the representation of 3D products. They came to the same conclusions that we have highlighted about the need of a better interoperability of geometric models and design systems with current DMUs. The proposed approaches made in this thesis with construction graph and morphological analysis of assembly offers new opportunities to adapt a model to the needs of the different applications involved in a product development process.

Bibliography

[AA96] Aichholzer O., Aurenhammer F.: Straight skeletons for general polygonal figures in the plane. In *Computing and Combinatorics*, Cai J.-Y., Wong C., (Eds.), vol. 1090. Springer Berlin Heidelberg, 1996, pp. 117–126. 57

[AAAG96] Aichholzer O., Aurenhammer F., Alberts D., Grtner B.: A novel type of skeleton for polygons. In *J.UCS The Journal of Universal Computer Science*, Maurer H., Calude C., Salomaa A., (Eds.). Springer Berlin Heidelberg, 1996, pp. 752–761. 57

[ABA02] Andujar C., Brunet P., Ayala D.: Topology-reducing surface simplification using a discrete solid representation. *ACM Trans. Graph.. Vol. 21*, Num. 2 (avril 2002), 88–105. 39, 44, 47, 64

[ABD*98] Armstrong C. G., Bridgett S. J., Donaghy R. J., Mccune R. W., Mckeag R. M., Robinson D. J.: Techniques for interactive and automatic idealisation of CAD models. In *Proceedings of the Sixth International Conference on Numerical Grid Generation in Computational Field Simulations* (Mississippi State, Mississippi, 39762 USA, 1998), NSF Engineering Research Center for Computational Field Simulation, ISGG, pp. 643–662. 35, 53

[ACK01] Amenta N., Choi S., Kolluri R. K.: The power crust, unions of balls, and the medial axis transform. *Computational Geometry. Vol. 19*, Num. 2 (2001), 127–153. 54

[AFS06] Attene M., Falcidieno B., Spagnuolo M.: Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer. Vol. 22*, Num. 3 (2006), 181–193. 59

[AKJ01] Anderson D., Kim Y. S., Joshi S.: A discourse on geometric feature recognition from cad models. *J Comput Inform Sci Eng. Vol. 1*, Num. 1 (2001), 440–746. 51

[AKM*06] Attene M., Katz S., Mortara M., Patané G., Spagnuolo M., Tal A.: Mesh segmentation-a comparative study. In *Shape*

Modeling and Applications, 2006. SMI 2006. IEEE International Conference on (2006), IEEE, pp. 7–7. 59

[ALC08] ALCAS: Advanced low-cost aircraft structures project. `http://alcas.twisoftware.com/`, 2005 – 2008. xv, 18

[AMP*02] Armstrong C. G., Monaghan D. J., Price M. A., Ou H., Lamont J.: Engineering computational technology. Civil-Comp press, Edinburgh, UK, UK, 2002, ch. Integrating CAE Concepts with CAD Geometry, pp. 75–104. 35

[Arm94] Armstrong C. G.: Modelling requirements for finite-element analysis. *Computer-aided design. Vol. 26*, Num. 7 (1994), 573–578. xvi, 48, 49, 132

[ARM*95] Armstrong C., Robinson D., McKeag R., Li T., Bridgett S., Donaghy R., McGleenan C.: Medials for meshing and more. In *Proceedings of the 4th International Meshing Roundtable* (1995). 48, 49, 53

[B*67] Blum H., et al.: A transformation for extracting new descriptors of shape. *Models for the perception of speech and visual form. Vol. 19*, Num. 5 (1967), 362–380. 48

[Bad11] Badin J.: *Ingénierie hautement productive et collaborative à base de connaissances métier: vers une méthodologie et un méta-modèle de gestion des connaissances en configurations.* PhD thesis, Belfort-Montbéliard, 2011. 44

[Bat96] Bathe K.-J.: *Finite element procedures*, vol. 2. Englewood Cliffs, NJ: Prentice-Hall, 1996. 24

[BBB00] Belaziz M., Bouras A., Brun J.-M.: Morphological analysis for product design. *Computer-Aided Design. Vol. 32*, Num. 5 (2000), 377–388. 63

[BBT08] Bellenger E., Benhafid Y., Troussier N.: Framework for controlled cost and quality of assumptions in finite element analysis. *Finite Elements in Analysis and Design. Vol. 45*, Num. 1 (2008), 25–36. 44

[BC04] Buchele S. F., Crawford R. H.: Three-dimensional halfspace constructive solid geometry tree construction from implicit boundary representations. *CAD. Vol. 36* (2004), 1063–1073. 62, 63, 107

[BCGM11] Badin J., Chamoret D., Gomes S., Monticolo D.: Knowledge configuration management for product design and numerical simulation. In *Proceedings of the 18th International Conference on Engineering Design (ICED11), Vol. 6* (2011), pp. 161–172. 44

[BCL12] Ba W., Cao L., Liu J.: Research on 3d medial axis transform via the saddle point programming method. *Computer-Aided Design. Vol. 44*, Num. 12 (2012), 1161 – 1172. 54

[BEGV08] Barequet G., Eppstein D., Goodrich M. T., Vaxman A.: Straight skeletons of three-dimensional polyhedra. In *Algorithms-ESA 2008*. Springer, 2008, pp. 148–160. 57

[Bic95] Bickford J.: *An Introduction to the Design and Behavior of Bolted Joints, Revised and Expanded*, vol. 97. CRC press, 1995. 178

[BKR*12] Barbau R., Krima S., Rachuri S., Narayanan A., Fiorentini X., Foufou S., Sriram R. D.: Ontostep: Enriching product model data using ontologies. *Computer-Aided Design. Vol. 44*, Num. 6 (2012), 575–590. 27

[BLHF12] Boussuge F., Léon J.-C., Hahmann S., Fine L.: An analysis of DMU transformation requirements for structural assembly simulations. In *The Eighth International Conference on Engineering Computational Technology* (Dubronik, Croatie, 2012), B.H.V. Topping, Civil Comp Press. 65, 82

[BLHF14a] Boussuge F., Lon J.-C., Hahmann S., Fine L.: Extraction of generative processes from b-rep shapes and application to idealization transformations. *Computer-Aided Design. Vol. 46*, Num. 0 (2014), 79 – 89. 2013 {SIAM} Conference on Geometric and Physical Modeling. 86

[BLHF14b] Boussuge F., Lon J.-C., Hahmann S., Fine L.: Idealized models for fea derived from generative modeling processes based on extrusion primitives. In *Proceedings of the 22nd International Meshing Roundtable* (2014), Sarrate J., Staten M., (Eds.), Springer International Publishing, pp. 127–145. 86

[BLNF07] Bellec J., Ladevèze P., Néron D., Florentin E.: Robust computation for stochastic problems with contacts. In *International Conference on Adaptive Modeling and Simulation* (2007). 82

[BR78] Babuvška I., Rheinboldt W. C.: Error estimates for adaptive finite element computations. *SIAM Journal on Numerical Analysis. Vol. 15*, Num. 4 (1978), 736–754. 46

[BS96] Butlin G., Stops C.: Cad data repair. In *Proceedings of the 5th International Meshing Roundtable* (1996), pp. 7–12. 47

[BSL*14] Boussuge F., Shahwan A., Lon J.-C., Hahmann S., Foucault G., Fine L.: Template-based geometric transformations of a functionally enriched dmu into fe assembly models. *Computer-Aided Design and Applications. Vol. 11*, Num. 4 (2014), 436–449. 168

[BWS03] Beall M. W., Walsh J., Shephard M. S.: Accessing cad geometry for mesh generation. In *IMR* (2003), pp. 33–42. 47

[CAS14] CASCADE O.: Open cascade technology, version 6.7.0 [computer software]. `http://www.opencascade.org/`, 1990 – 2014. 12, 112, 179, 186

[CAT14] CATIA D.: Dassault systèmes catia, version v6r2013x [computer software]. `http://www.3ds.com/products-services/catia/`, 2008 – 2014. 89, I, IX

[CBG08] Champaney L., Boucard P.-A., Guinard S.: Adaptive multi-analysis strategy for contact problems with friction. *Computational Mechanics. Vol. 42*, Num. 2 (2008), 305–315. 30, 175, 177

[CBL11] Cao L., Ba W., Liu J.: Computation of the medial axis of planar domains based on saddle point programming. *Computer-Aided Design. Vol. 43*, Num. 8 (2011), 979 – 988. 54

[Cha12] Champaney L.: A domain decomposition method for studying the effects of missing fasteners on the behavior of structural assemblies with contact and friction. *Computer Methods in Applied Mechanics and Engineering. Vol. 205* (2012), 121–129. 30, 175, 177

[CHE08] Clark B. W., Hanks B. W., Ernst C. D.: Conformal assembly meshing with tolerant imprinting. In *Proceedings of the 17th International Meshing Roundtable*. Springer, 2008, pp. 267–280. 65

[CP99] Chapman C., Pinfold M.: Design engineeringa need to rethink the solution using knowledge based engineering. *Knowledge-Based Systems. Vol. 12*, Num. 56 (1999), 257 – 267. 168

[CSKL04] Chong C., Senthil Kumar A., Lee K.: Automatic solid decomposition and reduction for non-manifold geometric model generation. *Computer-Aided Design. Vol. 36*, Num. 13 (2004), 1357–1369. 39, 44, 61, 95, 119

[CV06] Chouadria R., Veron P.: Identifying and re-meshing contact interfaces in a polyhedral assembly for digital mock-up. *Engineering with Computers. Vol. 22*, Num. 1 (2006), 47–58. 65

[DAP00] Donaghy R. J., Armstrong C. G., Price M. A.: Dimensional reduction of surface models for analysis. *Engineering with Computers. Vol. 16*, Num. 1 (2000), 24–35. 39, 45, 53

[DLG*07] Drieux G., Léon J.-C., Guillaume F., Chevassus N., Fine L., Poulat A.: Interfacing product views through a mixed shape representation. part 2: Model processing description. *International Journal on Interactive Design and Manufacturing (IJIDeM). Vol. 1*, Num. 2 (2007), 67–83. 43

[DMB*96] Donaghy R., McCune W., Bridgett S., Armstrong D., Robinson D., McKeag R.: Dimensional reduction of analysis models. In *Proceedings of the 5th International Meshing Roundtable* (1996). 53

[Dri06] Drieux G.: *De la maquette numrique produit vers ses applications aval : propositions de modles et procds associs.* PhD thesis, Institut National Polytechnique, Grenoble, FRANCE, 2006. 6, 19

[DSG97] Dey S., Shephard M. S., Georges M. K.: Elimination of the adverse effects of small model features by the local modification of automatically generated meshes. *Engineering with Computers. Vol. 13*, Num. 3 (1997), 134–152. 47

[Eck00] Eckard C.: Advantages and disavantadges of fem analysis in an early state of the design process. In *Proc. of the 2nd Worldwide Automotive Conference, MSC Software Corp, Dearborn, Michigan, USA* (2000). 44

[EF11] E. Florentin S.Guinard P. P.: A simple estimator for stress errors dedicated to large elastic finite element simulations: Locally reinforced stress construction. *Engineering Computations: Int J for Computer-Aided Engineering* (2011). 46

[FCF*08] Foucault G., Cuillière J.-C., François V., Léon J.-C., Maranzana R.: Adaptation of cad model topology for finite element analysis. *Computer-Aided Design. Vol. 40*, Num. 2 (2008), 176–196. xvi, 34, 50, 97

[FEA14] FEA A. U.: Dassault systèmes abaqus unified fea, version 6.13 [computer software]. `http://www.3ds.com/products-services/simulia/portfolio/abaqus/overview/`, 2005 – 2014. 180, XIX

[Fin01] FINE L.: *Processus et mthodes d'adaptation et d'idalisation de modles ddis l'analyse de structures mcaniques*. PhD thesis, Institut National Polytechnique, Grenoble, FRANCE, 2001. 22, 34, 45, 47

[FL*10] FOUCAULT G., LÉON J.-C., ET AL.: Enriching assembly cad models with functional and mechanical informations to ease cae. In *Proceedings of the ASME 2010 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2010 August 15-18, 2010, Montréal, Canada* (2010). 18, 20

[FLM03] FOSKEY M., LIN M. C., MANOCHA D.: Efficient computation of a simplified medial axis. In *Proceedings of the eighth ACM symposium on Solid modeling and applications* (2003), ACM, pp. 96–107. 54

[FMLG09] FERRANDES R., MARIN P. M., LÉON J. C., GIANNINI F.: A posteriori evaluation of simplification details for finite element model preparation. *Comput. Struct.. Vol. 87*, Num. 1-2 (janvier 2009), 73–80. 44

[Fou07] FOUCAULT G.: *Adaptation de modles CAO paramtrs en vue d'une analyse de comportement mcanique par lments finis*. PhD thesis, École de technologie supérieure, Montreal, CANADA, 2007. 13, 18, 46

[FRL00] FINE L., REMONDINI L., LEON J.-C.: Automated generation of fea models through idealization operators. *International Journal for Numerical Methods in Engineering. Vol. 49*, Num. 1-2 (2000), 83–108. 44

[GPu14] GPURE: Gpure, version 3.4 [computer software]. `http://www.gpure.net`, 2011 – 2014. 35

[GZL*10] GAO S., ZHAO W., LIN H., YANG F., CHEN X.: Feature suppression based cad mesh model simplification. *Computer-Aided Design. Vol. 42*, Num. 12 (2010), 1178–1188. 44

[HC03] HAIMES R., CRAWFORD C.: Unified geometry access for analysis and design. In *IMR* (2003), pp. 21–31. 47

[HLG*08] HAMRI O., LÉON J.-C., GIANNINI F., FALCIDIENO B., POULAT A., FINE L.: Interfacing product views through a mixed shape representation. part 1: Data structures and operators. *International Journal on Interactive Design and Manufacturing (IJIDeM). Vol. 2*, Num. 2 (2008), 69–85. 43

[HPR00] Han J., Pratt M., Regli W. C.: Manufacturing feature recognition from solid models: a status report. *Robotics and Automation, IEEE Transactions on. Vol. 16*, Num. 6 (2000), 782–796. 51

[HR84] Hoffman D. D., Richards W. A.: Parts of recognition. *Cognition. Vol. 18*, Num. 1 (1984), 65–96. 59

[HSKK01] Hilaga M., Shinagawa Y., Kohmura T., Kunii T. L.: Topology matching for fully automatic similarity estimation of 3d shapes. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM, pp. 203–212. 59

[Hut86] Huth H.: Influence of fastener flexibility on the prediction of load transfer and fatigue life for multiple-row joints. *Fatigue in mechanically fastened composite and metallic joints, ASTM STP. Vol. 927* (1986), 221–250. 29

[IIY*01] Inoue K., Itoh T., Yamada A., Furuhata T., Shimada K.: Face clustering of a large-scale cad model for surface mesh generation. *Computer-Aided Design. Vol. 33*, Num. 3 (2001), 251–261. 49

[IML08] Iacob R., Mitrouchev P., Léon J.-C.: Contact identification for assembly–disassembly simulation with a haptic device. *The Visual Computer. Vol. 24*, Num. 11 (2008), 973–979. 6

[ISO94] ISO:. ISO TC184-SC4: ISO-10303 Part 203 - Application Protocol: Configuration controlled 3D design of mechanical parts and assemblies, 1994. 33, 72, 89, 180, XIX

[ISO03] ISO:. ISO TC184-SC4: ISO-10303 Part 214 - Application Protocol: Core data for automotive mechanical design processes, 2003. 33, 72, 89, 180, XIX

[JB95] Jones M.R. M. P., Butlin G.: Geometry management support for auto-meshing. In *Proceedings of the 4th International Meshing Roundtable* (1995), pp. 153–164. 47

[JBH*14] Jourdes F., Bonneau G.-P., Hahmann S., Léon J.-C., Faure F.: Computation of components interfaces in highly complex assemblies. *Computer-Aided Design. Vol. 46* (2014), 170–178. xvi, 65, 71, 72, 79, 116

[JC88] Joshi S., Chang T.-C.: Graph-based heuristics for recognition of machined features from a 3d solid model. *Computer-Aided Design. Vol. 20*, Num. 2 (1988), 58–66. 51

[JD03] Joshi N., Dutta D.: Feature simplification techniques for freeform surface models. *Journal of Computing and Information Science in Engineering. Vol. 3* (2003), 177. 51

[JG00] Jha K., Gurumoorthy B.: Multiple feature interpretation across domains. *Computers in industry. Vol. 42*, Num. 1 (2000), 13–32. 51, 94, 100

[KLH*05] Kim S., Lee K., Hong T., Kim M., Jung M., Song Y.: An integrated approach to realize multi-resolution of b-rep model. In *Proceedings of the 2005 ACM symposium on Solid and physical modeling* (2005), ACM, pp. 153–162. 51

[Kos03] Koschan A.: Perception-based 3d triangle mesh segmentation using fast marching watersheds. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on* (2003), vol. 2, IEEE, pp. II–27. 59

[KT03] Katz S., Tal A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Trans. Graph.. Vol. 22*, Num. 3 (juillet 2003), 954–961. 59

[KWMN04] Kim K.-Y., Wang Y., Muogboh O. S., Nnaji B. O.: Design formalism for collaborative assembly design. *Computer-Aided Design. Vol. 36*, Num. 9 (2004), 849–871. 27, 64

[LAPL05] Lee K. Y., Armstrong C. G., Price M. A., Lamont J. H.: A small feature suppression/unsuppression system for preparing b-rep models for analysis. In *Proceedings of the 2005 ACM symposium on Solid and physical modeling* (New York, NY, USA, 2005), SPM '05, ACM, pp. 113–124. 35, 39, 44

[LDB05] Lavoué G., Dupont F., Baskurt A.: A new cad mesh segmentation method, based on curvature tensor analysis. *Computer-Aided Design. Vol. 37*, Num. 10 (2005), 975–987. 59

[Ley01] Leyton M.: *A Generative Theory of Shape (Lecture Notes in Computer Science, LNCS 2145)*. Springer-Verlag, 2001. 93

[LF05] Léon J.-C., Fine L.: A new approach to the preparation of models for fe analyses. *International journal of computer applications in technology. Vol. 23*, Num. 2 (2005), 166–184. 35, 39, 44, 45, 81

[LFLT14] Li K., Foucault G., Leon J.-C., Trlin M.: Fast global and partial reflective symmetry analyses using boundary surfaces of mechanical components. *Computer-Aided Design*, Num. 0 (2014), –. 194

[LG97] Liu S.-S., Gadh R.: Automatic hexahedral mesh generation by recursive convex and swept volume decomposition. In *Proceedings 6th International Meshing Roundtable, Sandia National Laboratories* (1997), Citeseer, pp. 217–231. 60

[LG05] Lockett H. L., Guenov M. D.: Graph-based feature recognition for injection moulding based on a mid-surface approach. *Computer-Aided Design. Vol. 37*, Num. 2 (2005), 251–262. 51

[LGT01] Lu Y., Gadh R., Tautges T. J.: Feature based hex meshing methodology: feature recognition and volume decomposition. *Computer-Aided Design. Vol. 33*, Num. 3 (2001), 221–232. 60, 100

[LL83] Ladeveze P., Leguillon D.: Error estimate procedure in the finite element method and applications. *SIAM Journal on Numerical Analysis. Vol. 20*, Num. 3 (1983), 485–509. 46

[LLKK04] Lee J. Y., Lee J.-H., Kim H., Kim H.-S.: A cellular topology-based approach to generating progressive solid models from feature-centric models. *Computer-Aided Design. Vol. 36*, Num. 3 (2004), 217–229. 51

[LLM06] Li M., Langbein F. C., Martin R. R.: Constructing regularity feature trees for solid models. In *Geometric Modeling and Processing-GMP 2006*. Springer, 2006, pp. 267–286. 63

[LLM10] Li M., Langbein F. C., Martin R. R.: Detecting design intent in approximate cad models using symmetry. *Computer-Aided Design. Vol. 42*, Num. 3 (2010), 183–201. 63

[LMTS*05] Lim T., Medellin H., Torres-Sanchez C., Corney J. R., Ritchie J. M., Davies J. B. C.: Edge-based identification of dp-features on free-form solids. *Pattern Analysis and Machine Intelligence, IEEE Transactions on. Vol. 27*, Num. 6 (2005), 851–860. 93

[LNP07a] Lee H., Nam Y.-Y., Park S.-W.: Graph-based midsurface extraction for finite element analysis. In *Computer Supported Cooperative Work in Design, 2007. CSCWD 2007. 11th International Conference on* (2007), pp. 1055–1058. 56

[LNP07b] Lee H., Nam Y.-Y., Park S.-W.: Graph-based midsurface extraction for finite element analysis. In *Computer Supported Cooperative Work in Design, 2007. CSCWD 2007. 11th International Conference on* (2007), pp. 1055–1058. 119

[LOC16] LOCOMACHS: Low cost manufacturing and assembly of composite and hybrid structures project. http://www.locomachs.eu/, 2012 – 2016. xv, 18

[LPA*03] Lee K., Price M., Armstrong C., Larson M., Samuelsson K.: Cad-to-cae integration through automated model simplification and adaptive modelling. In *International Conference on Adaptive Modeling and Simulation* (2003). 47, 49

[LPMV10] Lou R., Pernot J.-P., Mikchevitch A., Véron P.: Merging enriched finite element triangle meshes for fast prototyping of alternate solutions in the context of industrial maintenance. *Computer-Aided Design. Vol. 42*, Num. 8 (2010), 670–681. 65

[LSJS13] Lee-St John A., Sidman J.: Combinatorics and the rigidity of cad systems. *Computer-Aided Design. Vol. 45*, Num. 2 (2013), 473–482. 19

[LST*12] Li K., Shahwan A., Trlin M., Foucault G., Léon J.-C.: Automated contextual annotation of b-rep cad mechanical components deriving technology and symmetry information to support partial retrieval. In *Proceedings of the 5th Eurographics Conference on 3D Object Retrieval* (Aire-la-Ville, Switzerland, Switzerland, 2012), EG 3DOR'12, Eurographics Association, pp. 67–70. 20

[LZ04] Liu R., Zhang H.: Segmentation of 3d meshes through spectral clustering. In *Computer Graphics and Applications, 2004. PG 2004. Proceedings. 12th Pacific Conference on* (2004), IEEE, pp. 298–305. 59

[Man88] Mantyla M.: *An Introduction to Solid Modeling*. W. H. Freeman & Co., New York, NY, USA, 1988. 93

[MAR12] Makem J., Armstrong C., Robinson T.: Automatic decomposition and efficient semi-structured meshing of complex solids. In *Proceedings of the 20th International Meshing Roundtable*, Quadros W., (Ed.). Springer Berlin Heidelberg, 2012, pp. 199–215. xvi, 60, 119, 126

[MCC98] Mobley A. V., Carroll M. P., Canann S. A.: An object oriented approach to geometry defeaturing for finite element meshing. In *Proceedings of the 7th International Meshing Roundtable, Sandia National Labs* (1998), pp. 547–563. 35

[MCD11] Musuvathy S., Cohen E., Damon J.: Computing medial axes of generic 3d regions bounded by b-spline surfaces. *Computer-Aided*

Design. Vol. 43, Num. 11 (2011), 1485 – 1495. ¡ce:title¿Solid and Physical Modeling 2011¡/ce:title¿. 54

[MGP10] Miklos B., Giesen J., Pauly M.: Discrete scale axis representations for 3d geometry. *ACM Transactions on Graphics (TOG). Vol. 29*, Num. 4 (2010), 101. 54

[OH04] O. Hamri J-C. Leon F. G.: A new approach of interoperability between cad and simulation models. In *TMCE* (2004). 48

[PFNO98] Peak R. S., Fulton R. E., Nishigaki I., Okamoto N.: Integrating engineering design and analysis using a multi-representation approach. *Engineering with Computers. Vol. 14*, Num. 2 (1998), 93–114. 44

[QO12] Quadros W. R., Owen S. J.: Defeaturing cad models using a geometry-based size field and facet-based reduction operators. *Engineering with Computers. Vol. 28*, Num. 3 (2012), 211–224. 47

[QVB*10] Quadros W., Vyas V., Brewer M., Owen S., Shimada K.: A computational framework for automating generation of sizing function in assembly meshing via disconnected skeletons. *Engineering with Computers. Vol. 26*, Num. 3 (2010), 231–247. 65

[RAF11] Robinson T., Armstrong C., Fairey R.: Automated mixed dimensional modelling from 2d and 3d cad models. *Finite Elements in Analysis and Design. Vol. 47*, Num. 2 (2011), 151 – 165. xvi, 44, 53, 54, 81, 119

[RAM*06] Robinson T. T., Armstrong C. G., McSparron G., Quenardel A., Ou H., McKeag R. M.: Automated mixed dimensional modelling for the finite element analysis of swept and revolved cad features. In *Proceedings of the 2006 ACM symposium on Solid and physical modeling* (2006), SPM '06, pp. 117–128. xvi, 53, 60, 61, 87, 126, 146

[RBO02] Rib R., Bugeda G., Oate E.: Some algorithms to correct a geometry in order to create a finite element mesh. *Computers and Structures. Vol. 80*, Num. 1617 (2002), 1399 – 1408. 47

[RDCG12] Russ B., Dabbeeru M. M., Chorney A. S., Gupta S. K.: Automated assembly model simplification for finite element analysis. In *ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (2012), American Society of Mechanical Engineers, pp. 197–206. 64

[Req77] Requicha A.: *Mathematical models of rigid solid objects.* Tech. rep., Technical Memorandum - 28, Rochester Univ., N.Y. Production Automation Project., 1977. 8

[Req80] Requicha A. G.: Representations for rigid solids: Theory, methods, and systems. *ACM Computing Surveys (CSUR). Vol. 12*, Num. 4 (1980), 437–464. 8

[Rez96] Rezayat M.: Midsurface abstraction from 3d solid models: general theory and applications. *Computer-Aided Design. Vol. 28*, Num. 11 (1996), 905 – 915. xvi, 35, 55, 56, 95, 100, 134

[RG03] Ramanathan M., Gurumoorthy B.: Constructing medial axis transform of planar domains with curved boundaries. *Computer-Aided Design. Vol. 35*, Num. 7 (2003), 619 – 632. 39, 54

[RG04] Ramanathan M., Gurumoorthy B.: Generating the mid-surface of a solid using 2d mat of its faces. *Computer-Aided Design and Applications. Vol. 1*, Num. 1-4 (2004), 665–674. 56

[RG10] Ramanathan M., Gurumoorthy B.: Interior medial axis transform computation of 3d objects bound by free-form surfaces. *Computer-Aided Design. Vol. 42*, Num. 12 (2010), 1217 – 1231. 54

[Roc12] Rocca G. L.: Knowledge based engineering: Between {AI} and cad. review of a language based technology to support engineering design. *Advanced Engineering Informatics. Vol. 26*, Num. 2 (2012), 159 – 179. Knowledge based engineering to support complex product design. 168

[ROM14] ROMMA: Robust mechanical models for assemblies. `http://romma.lmt.ens-cachan.fr/`, 2010 – 2014. xix, 177, 180, 189

[Sak95] Sakurai H.: Volume decomposition and feature recognition: Part 1polyhedral objects. *Computer-Aided Design. Vol. 27*, Num. 11 (1995), 833–843. 61

[SBBC00] Sheffer A., Bercovier M., Blacker T., Clements J.: Virtual topology operators for meshing. *International Journal of Computational Geometry & Applications. Vol. 10*, Num. 03 (2000), 309–331. 49

[SBO98] Shephard M. S., Beall M. W., O'Bara R. M.: Revisiting the elimination of the adverse effects of small model features in automatically generated meshes. In *IMR* (1998), pp. 119–131. 47

[SD96] Sakurai H., Dave P.: Volume decomposition and feature recognition, part ii: curved objects. *Computer-Aided Design. Vol. 28*, Num. 6 (1996), 519–537. 61

[SGZ10] Sun R., Gao S., Zhao W.: An approach to b-rep model simplification based on region suppression. *Computers & Graphics. Vol. 34*, Num. 5 (2010), 556–564. 39

[Sha95] Shah J. J.: *Parametric and feature-based CAD/CAM: concepts, techniques, and applications*. John Wiley & Sons, 1995. 13, 50

[She01] Sheffer A.: Model simplification for meshing using face clustering. *Computer-Aided Design. Vol. 33*, Num. 13 (2001), 925–934. 34, 49

[Sil81] Silva C. E.: *Alternative definitions of faces in boundary representatives of solid objects*. Tech. rep., Tech. Memo. 36, Production Automation Project, Univ. of Rochester, Rochester, N.Y., 1981, 1981. 97

[SLF*12] Shahwan A., Léon J.-C., Fine L., Foucault G., et al.: Reasoning about functional properties of components based on geometrical descriptions. In *Proceedings of the ninth International Symposium on Tools and Methods of Competitive Engineering, Karlsruhe, Germany)* (2012), TMCE12. 38, 71, 73, 79, 116, 168, 169

[SLF*13] Shahwan A., Léon J.-C., Foucault G., Trlin M., Palombi O.: Qualitative behavioral reasoning from components interfaces to components functions for dmu adaption to fe analyses. *Computer-Aided Design. Vol. 45*, Num. 2 (2013), 383–394. 19, 20, 27, 38, 71, 72, 73, 79, 116, 142, 160, 168, 169, 179

[SRX07] Stroud I., Renner G., Xirouchakis P.: A divide and conquer algorithm for medial surface calculation of planar polyhedra. *Computer-Aided Design. Vol. 39*, Num. 9 (2007), 794–817. 44, 54

[SS14] Shapiro V., Srinivasan V.: Opinion: Report from a 2013 asme panel on geometric interoperability for advanced manufacturing. *Comput. Aided Des.. Vol. 47* (fvrier 2014), A1–A2. 196

[SSCO08] Shapira L., Shamir A., Cohen-Or D.: Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer. Vol. 24*, Num. 4 (2008), 249–259. 59

[SSK*05] Seo J., Song Y., Kim S., Lee K., Choi Y., Chae S.: Wrap-around operation for multi-resolution cad model. *Computer-Aided Design & Applications. Vol. 2*, Num. 1-4 (2005), 67–76. 51

[SSM*10] Sheen D.-P., Son T.-g., Myung D.-K., Ryu C., Lee S. H., Lee K., Yeo T. J.: Transformation of a thin-walled solid model into a surface model via solid deflation. *Comput. Aided Des.. Vol. 42*, Num. 8 (aot 2010), 720–730. 39, 44, 57, 95, 100, 119, 126, 134

[SSR*07] Sheen D.-P., Son T.-g., Ryu C., Lee S. H., Lee K.: Dimension reduction of solid models by mid-surface generation. *International Journal of CAD/CAM. Vol. 7*, Num. 1 (2007). 57

[Str10] Stroud I.: *Boundary Representation Modelling Techniques*, 1st ed. Springer Publishing Company, Incorporated, 2010. 10

[SV93] Shapiro V., Vossler D. L.: Separation for boundary to csg conversion. *ACM Trans. Graph.. Vol. 12*, Num. 1 (1993), 35–55. 62, 63, 107

[Sza96] Szabo B.: The problem of model selection in numerical simulation. *Advances in computational methods for simulation* (1996), 9–16. 22

[Tau01] Tautges T. J.: Automatic detail reduction for mesh generation applications. In *Proceedings 10th International Meshing Roundtable* (2001), pp. 407–418. 35, 51

[TBG09] Thakur A., Banerjee A. G., Gupta S. K.: A survey of {CAD} model simplification techniques for physics-based simulation applications. *Computer-Aided Design. Vol. 41*, Num. 2 (2009), 65 – 80. 39, 51

[TNRA14] Tierney C. M., Nolan D. C., Robinson T. T., Armstrong C. G.: Managing equivalent representations of design and analysis models. *Computer-Aided Design and Applications. Vol. 11*, Num. 2 (2014), 193–205. 12

[Tro99] Troussier N.: *Contribution to the integration of mechanical calculation in engineering design : methodological proposition for the use and the reuse.* PhD thesis, University Joseph Fourier, Grenoble, FRANCE, 1999. 29, 44, 76

[TWKW59] Timoshenko S., Woinowsky-Krieger S., Woinowsky S.: *Theory of plates and shells*, vol. 2. McGraw-hill New York, 1959. 25, 126

[Ull09] Ullman D. G.: *The mechanical design process*, vol. Fourth Edition. McGraw-Hill Science/Engineering/Math, 2009. 175

[vdBvdMB03] van den Berg E., van der Meiden H. A., Bronsvoort W. F.: Specification of freeform features. In *Proceedings of the Eighth ACM*

Symposium on Solid Modeling and Applications (New York, NY, USA, 2003), SM '03, ACM, pp. 56–64. 173

[VR93] Vandenbrande J. H., Requicha A. A.: Spatial reasoning for the automatic recognition of machinable features in solid models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on. Vol. 15*, Num. 12 (1993), 1269–1285. 51

[VSR02] Venkataraman S., Sohoni M., Rajadhyaksha R.: Removal of blends from boundary representation models. In *Proceedings of the seventh ACM symposium on Solid modeling and applications* (2002), ACM, pp. 83–94. 35, 52, 93

[Woo03] Woo Y.: Fast cell-based decomposition and applications to solid modeling. *Computer-Aided Design. Vol. 35*, Num. 11 (2003), 969–977. 51, 61, 94

[Woo14] Woo Y.: Abstraction of mid-surfaces from solid models of thin-walled parts: A divide-and-conquer approach. *Computer-Aided Design. Vol. 47* (2014), 1–11. xvi, 44, 61, 62, 89, 94, 119

[WS02] Woo Y., Sakurai H.: Recognition of maximal features by volume decomposition. *Computer-Aided Design. Vol. 34*, Num. 3 (2002), 195–207. 51, 61, 94, 100

[ZM02] Zhu H., Menq C.: B-rep model simplification by automatic fillet/round suppressing for efficient automatic feature recognition. *Computer-Aided Design. Vol. 34*, Num. 2 (2002), 109–123. 35, 52, 93

[ZPK*02] Zhang Y., Paik J., Koschan A., Abidi M. A., Gorsich D.: Simple and efficient algorithm for part decomposition of 3-d triangulated models based on curvature analysis. In *Image Processing. 2002. Proceedings. 2002 International Conference on* (2002), vol. 3, IEEE, pp. III–273. 59

[ZT00] Zienkiewicz O. C., Taylor R. L.: *The Finite Element Method: Solid Mechanics*, vol. 2. Butterworth-heinemann, 2000. 24

Appendix A

Illustration of generation processes of CAD components

This appendix shows the construction process of two industrial use-cases. The components have been designed in CATIA [CAT14] CAD software.

A.1 Construction process of an injected plastic part

The followings figures present the complete shape generation process of the use-case as shown in Figure 4.1. The component has been designed with the successive application of 37 modeling features. The used features are:

1. Material addition or removal;
2. Surface operations: fillets, chamfers.

Two views, defined as top and bottom view, are associated with the construction tree to present all modeling step of the object shape.

A.2 Construction process of an aeronautical metallic part

The following figures present a complete shape generation of a simple metallic component which is commonly found in aeronautical structures. The component has been designed using the successive application of boolean operations of type addition and removal. This is a common practice for aeronautical metallic design. This technique shows directly the machining steps in the component design but turn the shape generation process quite complex. The simple example presented in Figure A.6 contains nine main operations which could be reduced to three majors operations (one extrusion and two hole drillings).

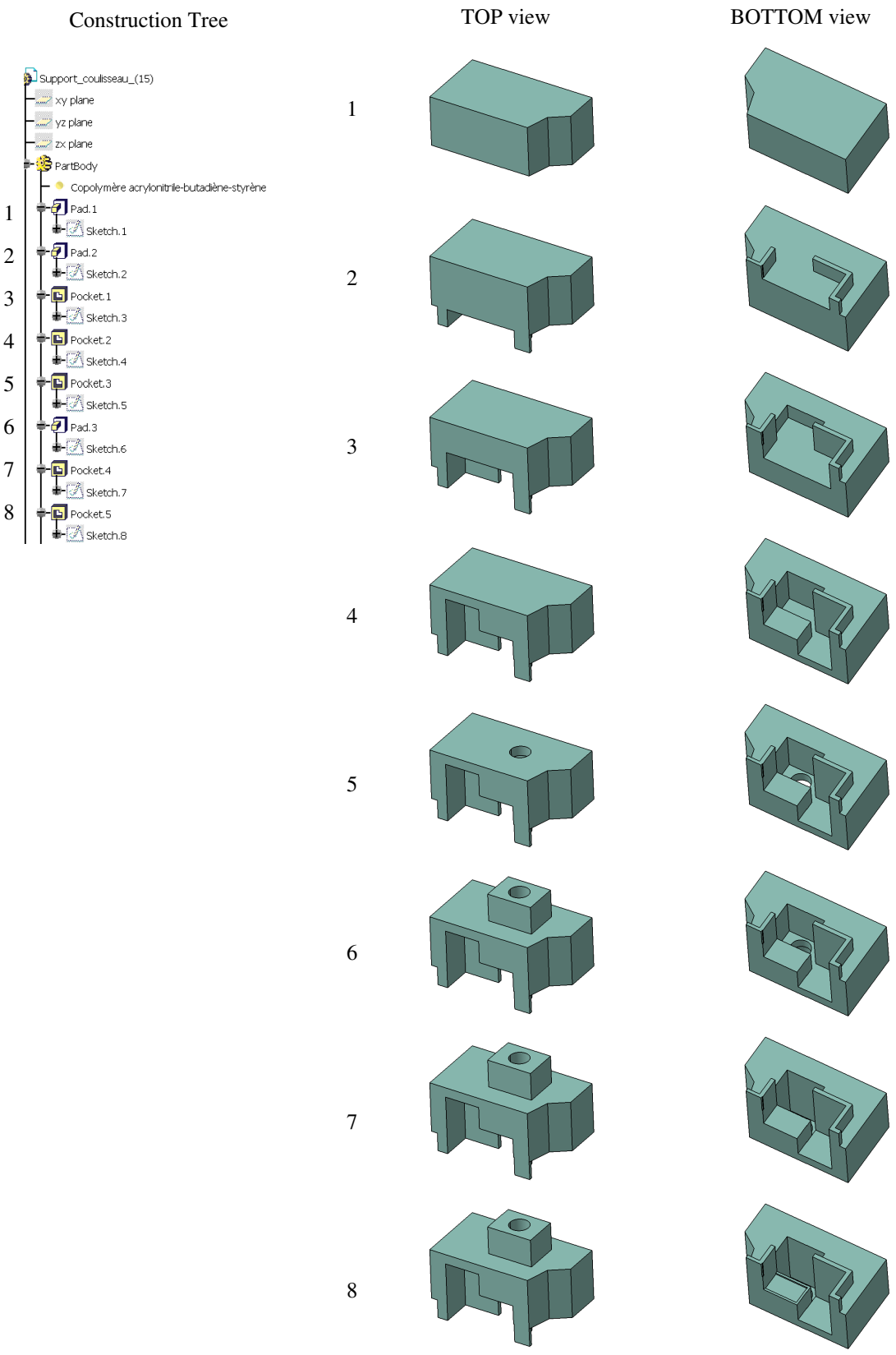


Figure A.1: An example of a shape generation process of an injected plastic part 1/5

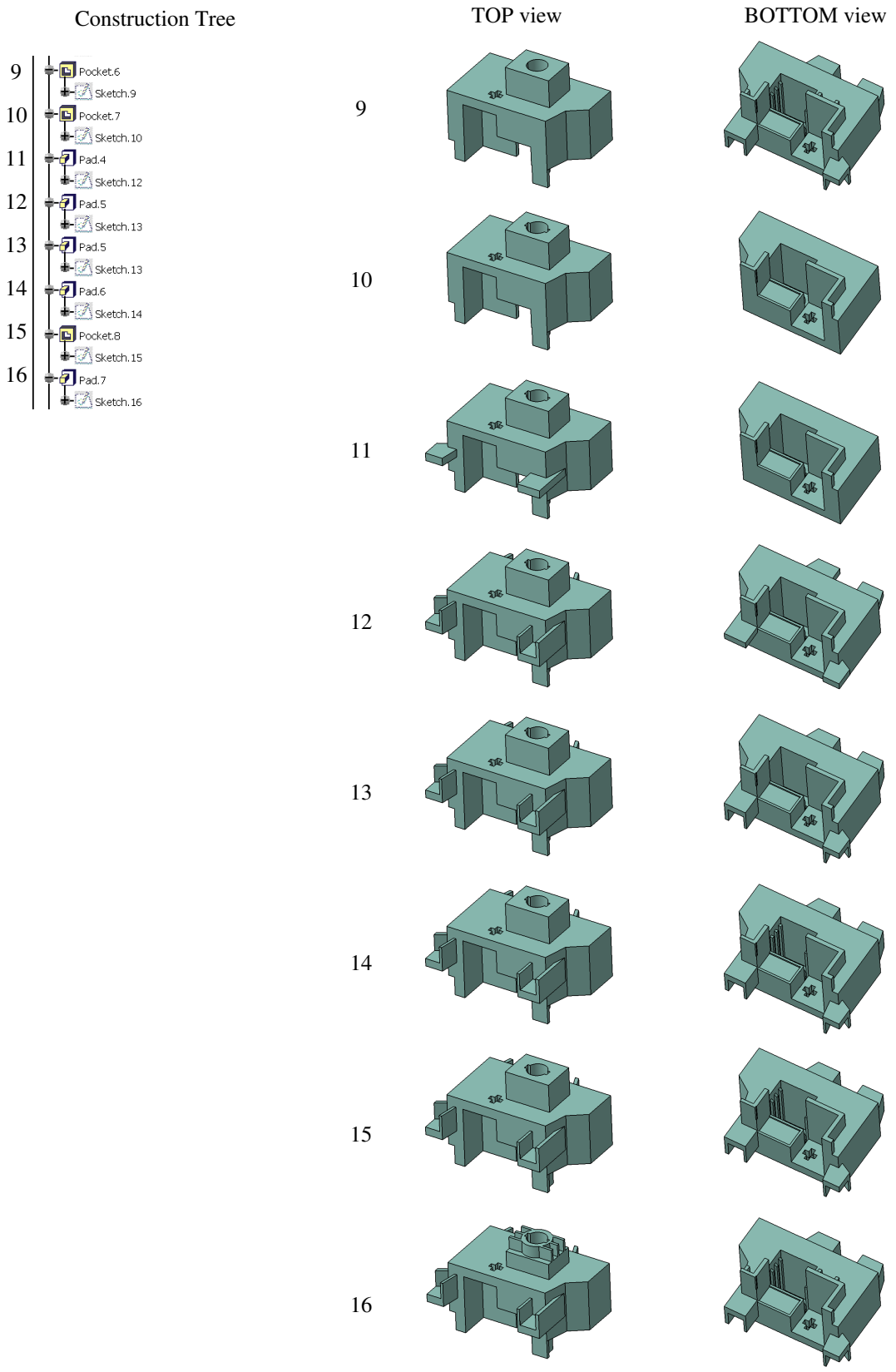


Figure A.2: An example of a shape generation process of an injected plastic part 2/5

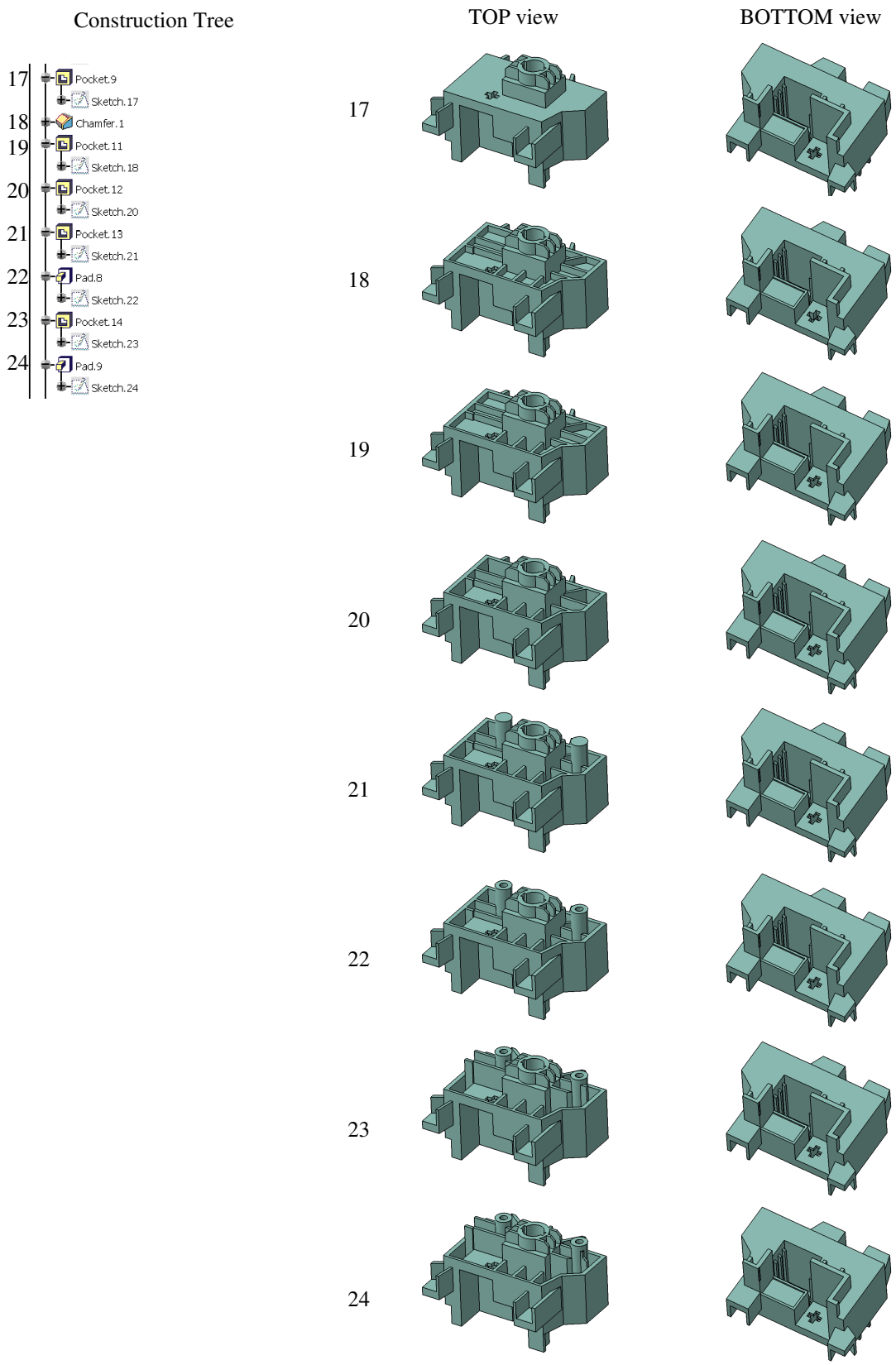


Figure A.3: An example of a shape generation process of an injected plastic part 3/5

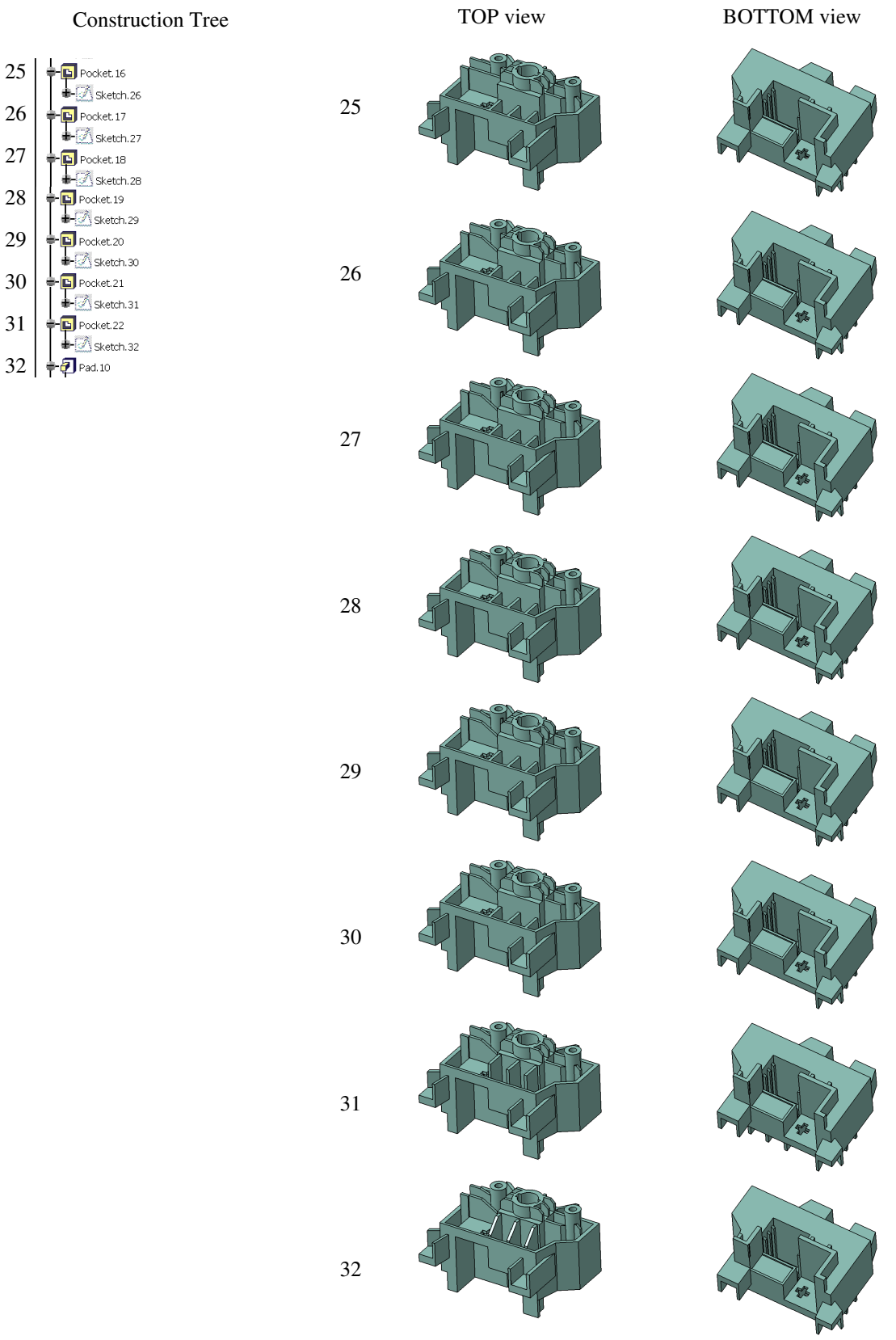


Figure A.4: An example of a shape generation process of an injected plastic part 4/5

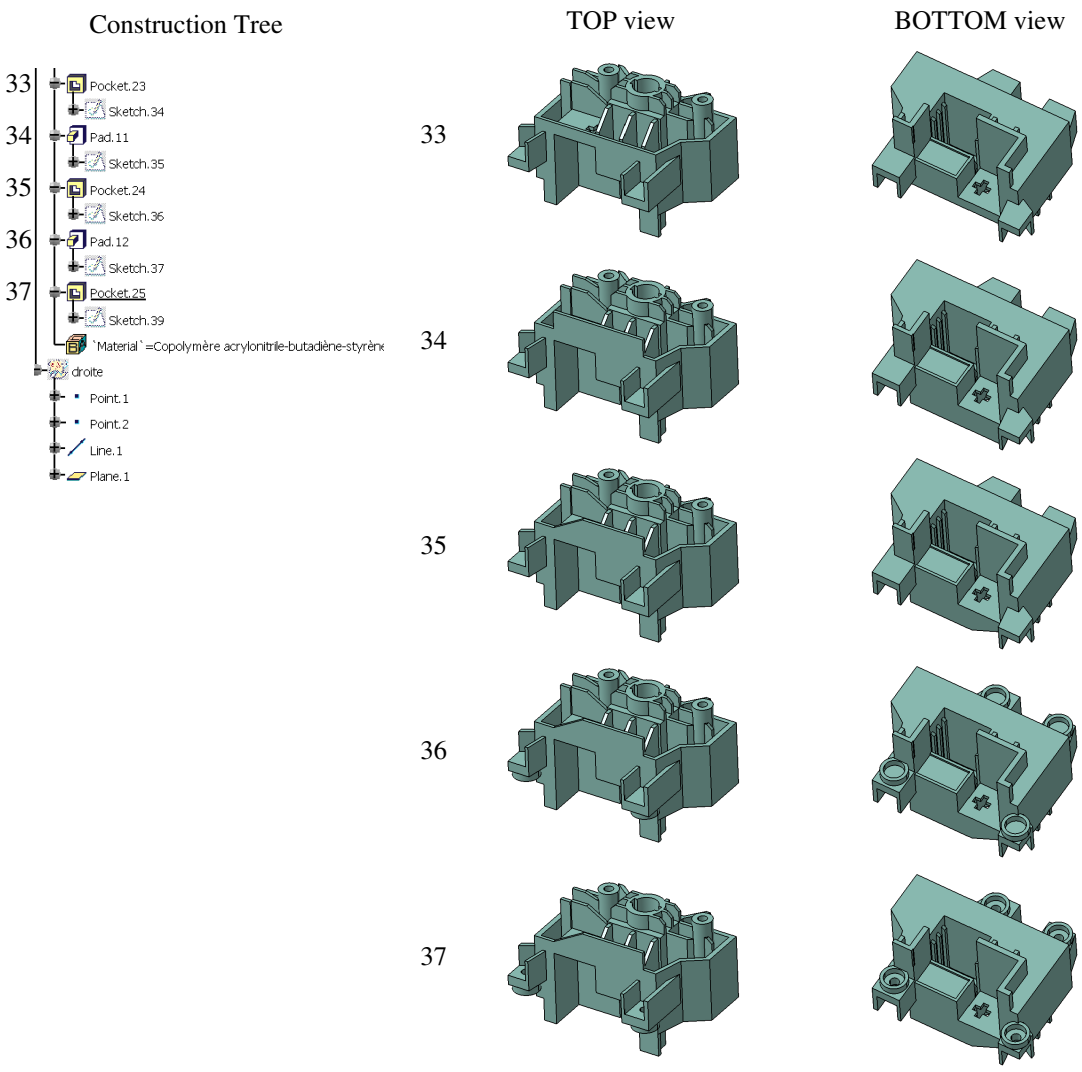


Figure A.5: An example of a shape generation process of an injected plastic part 5/5

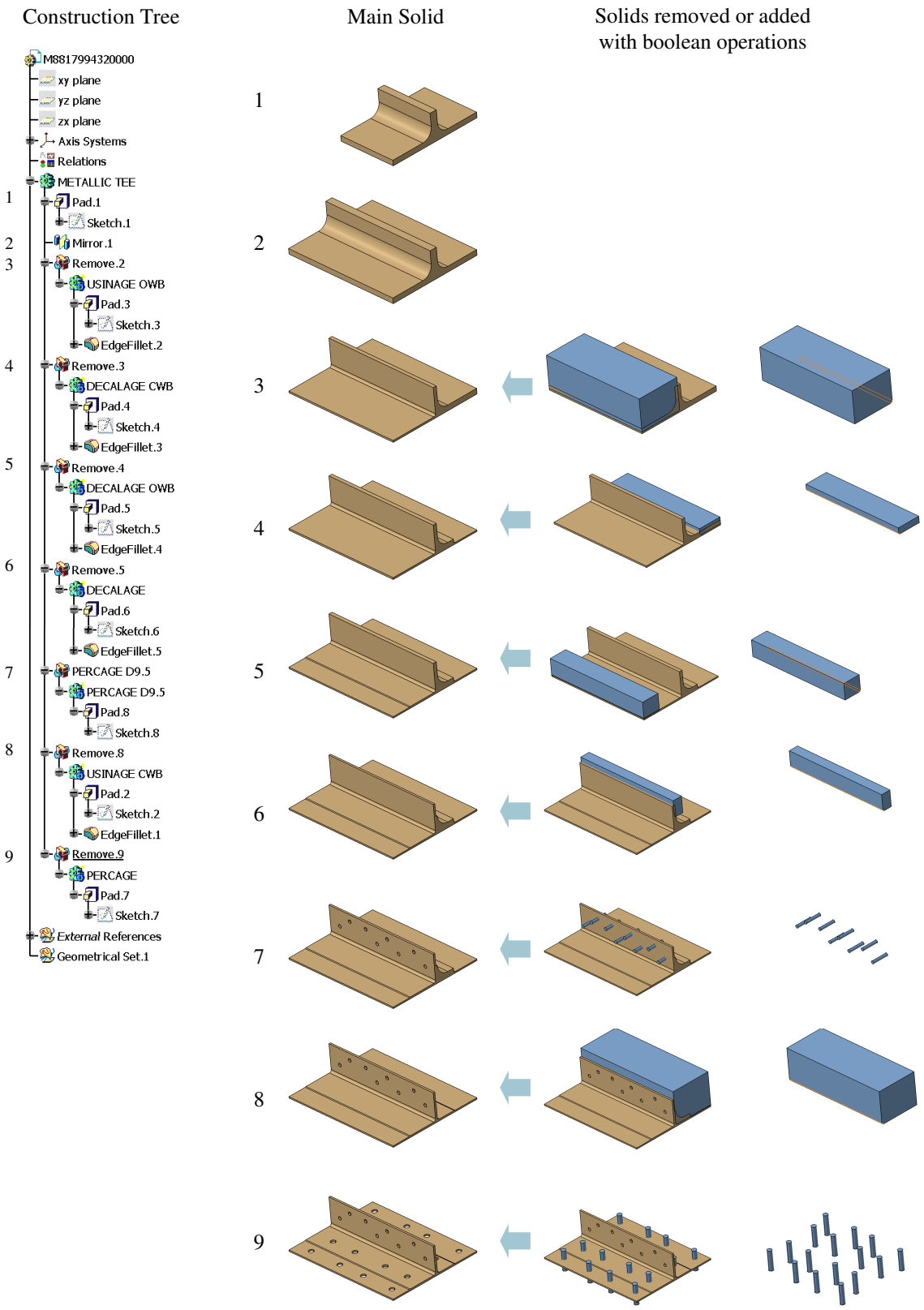


Figure A.6: An example of a shape generation process of a simple metallic component. The component has been mainly designed using boolean operations.

Appendix B

Features equivalence

This appendix illustrates the main modeling features used in CAD software to design components (Illustrations courtesy of Dassault Systèmes [CAT14])

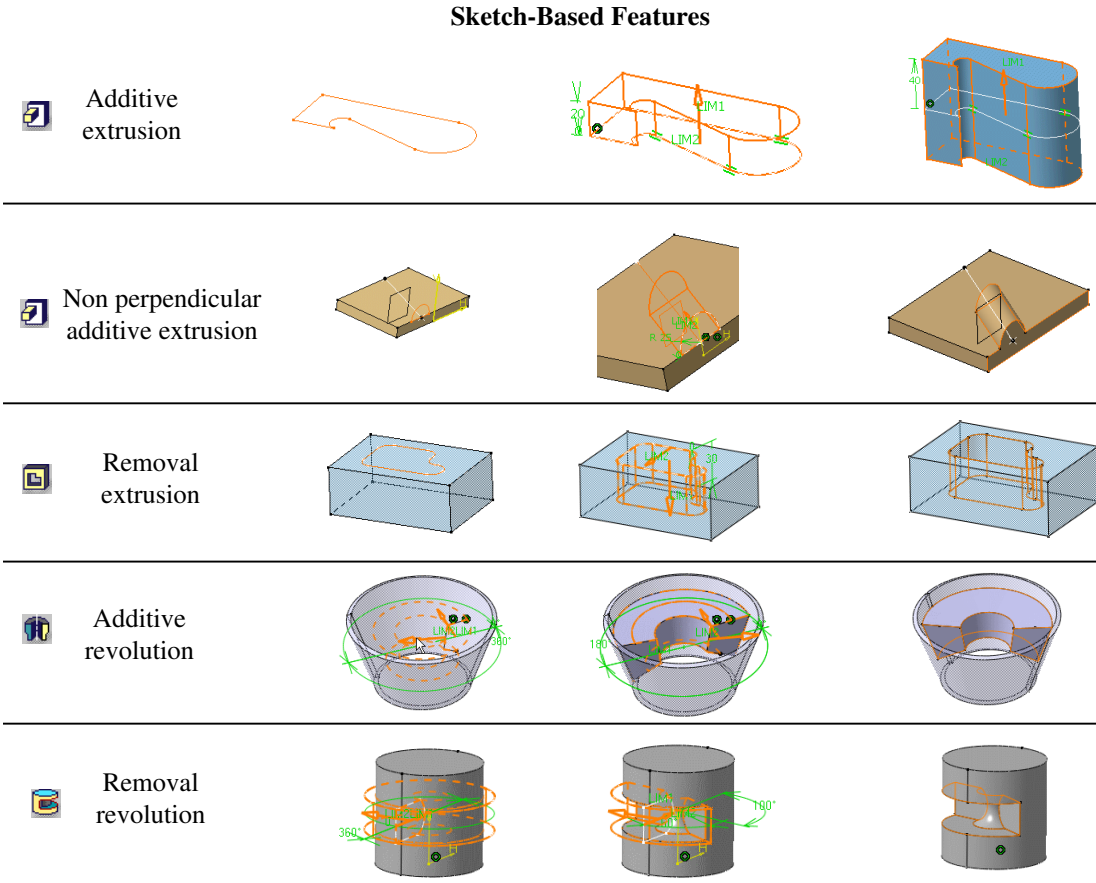
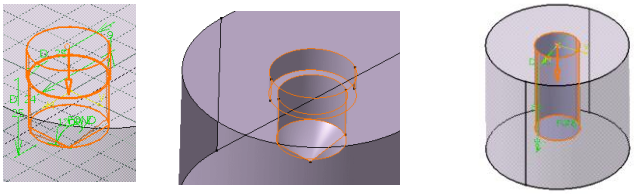


Figure B.1: Examples of Sketch-Based Features

Sketch-Based Features

Hole Drilling

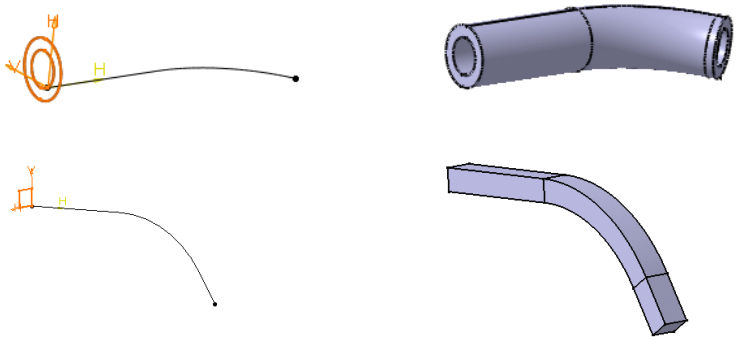


Hole type:

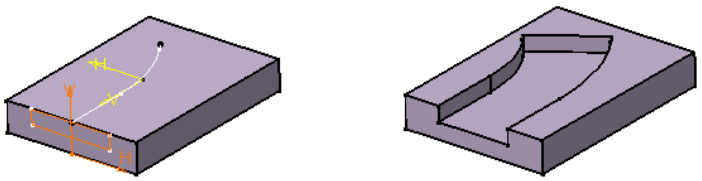
Equivalent to:

1 removal extrusion or 1 removal revolution	1 removal revolution	2 Removal extrusion or 1 removal revolution	1 removal revolution	1 removal revolution

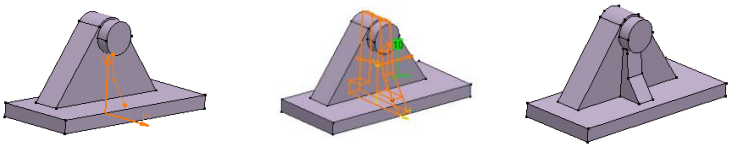
Additive sweep



Removal sweep



Stiffener

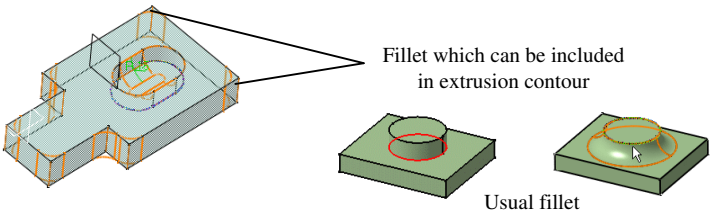


Equivalent to additive extrusion

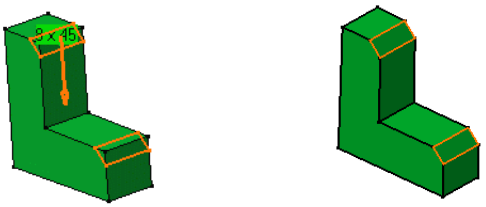
Figure B.2: Examples of Sketch-Based Features

Dress-Up Features

Fillet

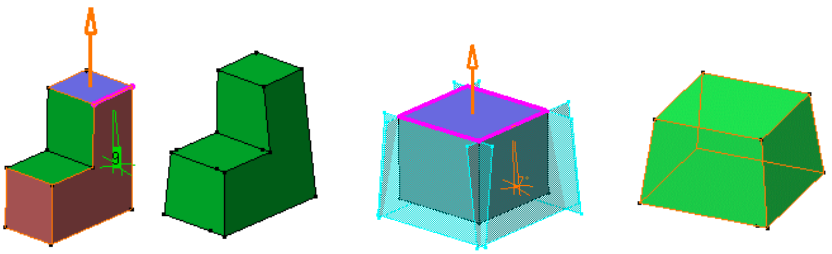


Chamfer

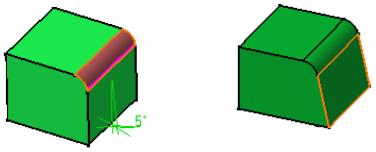


Equivalent to removal extrusion of can be included in additive extrusion contour

Draft angle

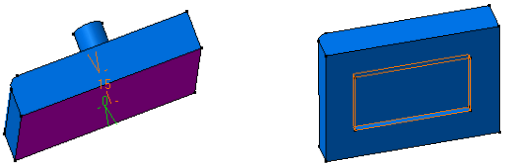


Draft angle feature can be equivalent to various additive and removal extrusions



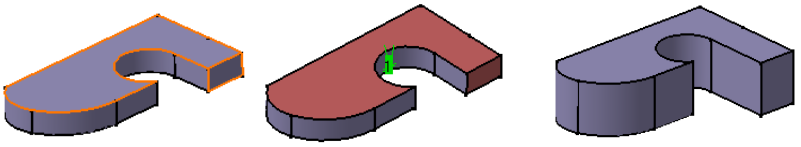
Simple draft can be included in additive extrusion contour

Shell



Equivalent to removal extrusion

Thickness



Equivalent to additive extrusion

Figure B.3: Examples of Dress-Up Features

Boolean operations

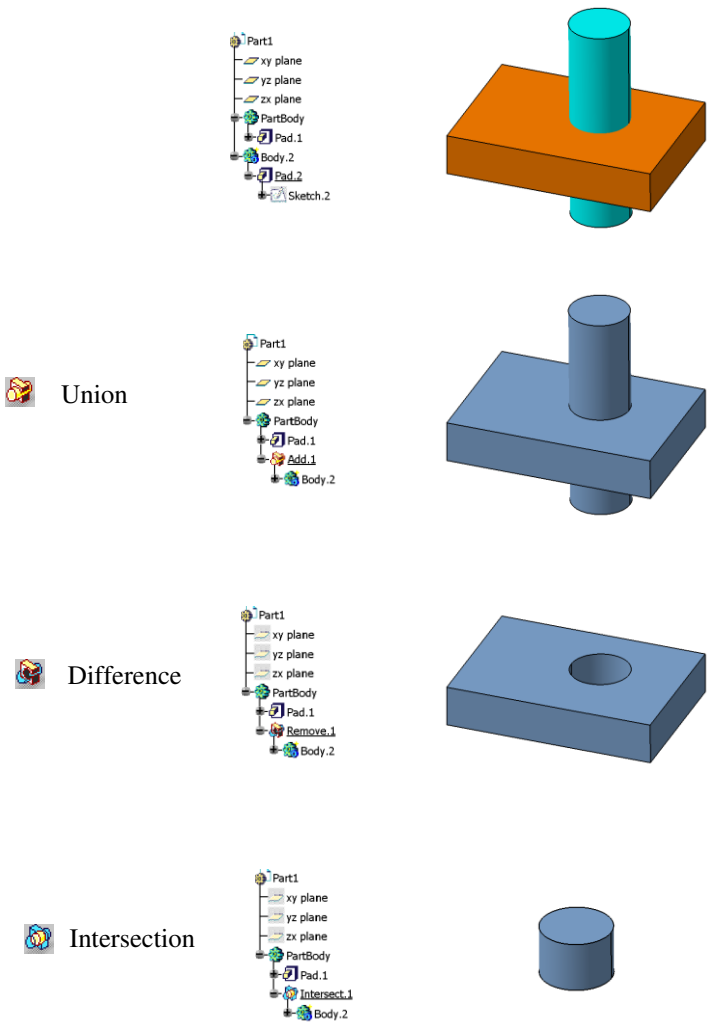


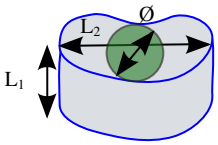
Figure B.4: Examples of Boolean operations

Appendix C

Taxonomy of a primitive morphology

This appendix illustrates 18 morphological configurations associated with a MAT medial edge of a volume primitive P_i of type extrusion.

The two tables differ according to whether the idealization direction of P_i corresponds to the extrusion direction, see Table C.1, or whether the idealization direction of P_i is included in the extrusion contour, see Table C.2. The reference ratio x_r and user ratio x_u are used to specify, in each table, the intervals of morphology differentiating beams, plates or shells and 3D thick domains.



x = max Ø / L_1 if max Ø > L_1
x = L_1 / max Ø if max Ø <= L_1
y = L_2 / max Ø

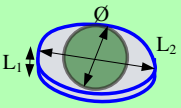
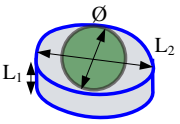
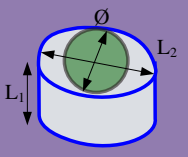
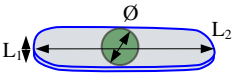
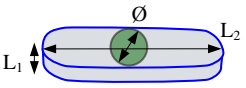
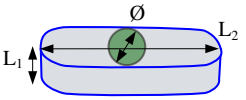
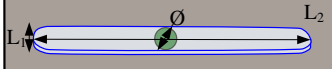
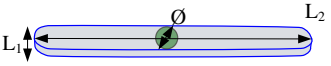
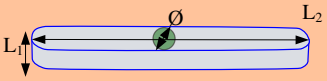
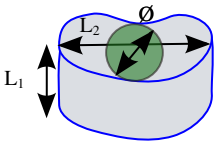
x = max Ø / L_1 y = L2 / max Ø	x_r < x	x_r < x < x_u	x < x_u
y < xu	L_1 < max Ø / x_r max Ø < L_2 < x_u . max Ø  PLATE	max Ø / x_r < L_1 < max Ø / x_u max Ø < L_2 < x_u . max Ø  PLATE under user hypothesis (in plane Ø)	x_u . max Ø < L_1 < max Ø max Ø < L_2 < x_u . max Ø  THICK
xu < y < x_r	L1 < max Ø / x_r x_u . max Ø < L_2 < x_r . max Ø  BEAM to BAND under user hypothesis	max Ø / x_r < L1 < max Ø / x_u x_u . max Ø < L_2 < x_r . max Ø  BEAM to PLATE under user hypothesis	x_u . max Ø < L1 < max Ø x_u . max Ø < L2 < x_r . max Ø  BEAM under user hypothesis (direction L_2)
x_r < y	L1 < max Ø / x_r x_u . max Ø < L_2  BAND	max Ø / x_r < L1 < max Ø / x_u x_u . max Ø < L_2  PLATE (plane orthogonal Ø) to BAND under user hypothesis	x_u . max Ø < L1 < max Ø x_u . max O < L_2  BEAM (direction L_2)

Table C.1: Morphology associated with a MAT medial edge of a primitive P_i. 1/2



x = max Ø / L_1 if max Ø > L_1
x = L_1 / max Ø if max Ø <= L_1
y = L_2 / max Ø

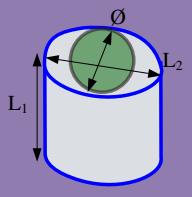
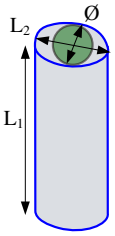
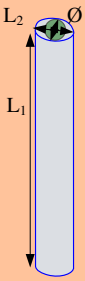
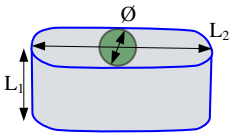
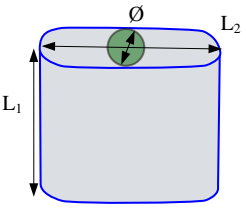
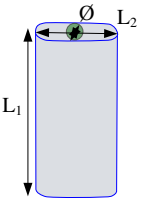
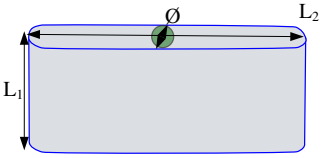
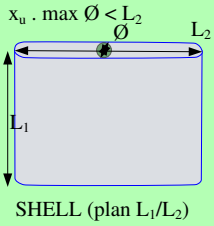
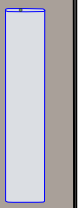
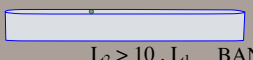
x = max Ø / L_1 y = L2 / max Ø	x < x_u	x_u < x < x_r	x_r < x
y < xu	max Ø < L_1 < x_u . max Ø max Ø < L_2 < x_u . max Ø  MASSIF	x_u . max Ø < L_1 < x_r . max Ø max Ø < L_2 < x_u . max Ø  BEAM under user hypothesis (direction L_1)	x_r . max Ø < L_1 max Ø < L_2 < x_u . max Ø  BEAM (direction L_1)
xu < y < x_r	max Ø < L_1 < x_u . max Ø x_u . max Ø < L_2 < x_r . max Ø  SHELL under user hypothesis (plane L_1/L_2)	x_u . max Ø < L_1 < x_r . max Ø x_u . max Ø < L_2 < x_r . max Ø  SHELL under user hypothesis (plane L_1/L_2)	x_r . max Ø < L_1 x_u . max Ø < L_2 < x_r . max Ø  BEAM (direction L_1) to SHELL under user hypothesis (plane L_1/L_2)  L_1 > 10 . L_2 BEAM
x_r < y	max Ø < L_1 < x_u . max Ø x_u . max Ø < L_2  BEAM (direction L_2)	x_u . max Ø < L_1 < x_r . max Ø x_u . max Ø < L_2  SHELL under user hypothesis (plane L_1/L_2)	x_r . max Ø < L_1 x_u . max Ø < L_2  SHELL (plan L_1/L_2)  L_1 > 10 . L_2 BAND  L_2 > 10 . L_1 BAND

Table C.2: Morphology associated with a MAT medial edge of a primitive P_i. 2/2

Appendix D

Export to CAE software

This appendix illustrates the data structure used to transfer the adapted DMU to a CAE software, i.e., Abaqus [FEA14]. STEP files [ISO94, ISO03] are used to transfer the geometric model and associated xml files.

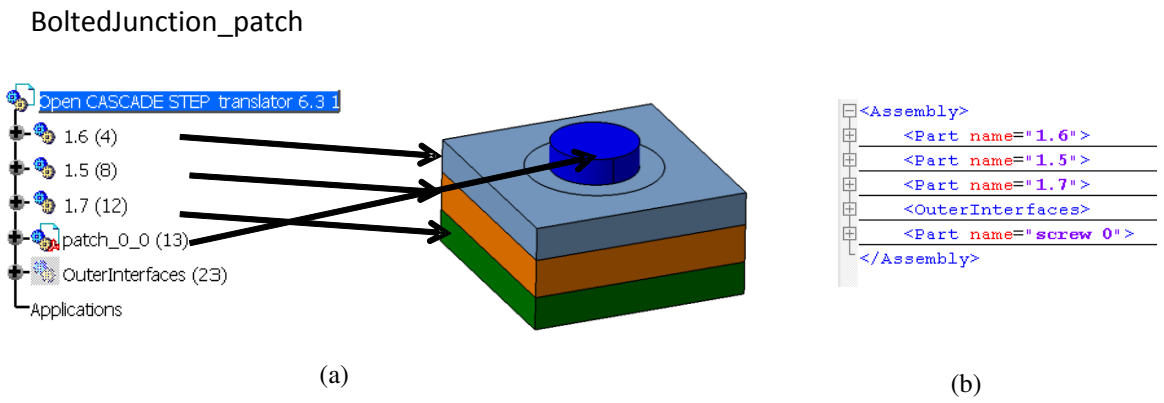


Figure D.1: Illustration of the STEP export of a Bolted Junction with sub-domains around screw. (a) Product structure open in CATIA software, (b) associated xml file containing the association between components and interfaces

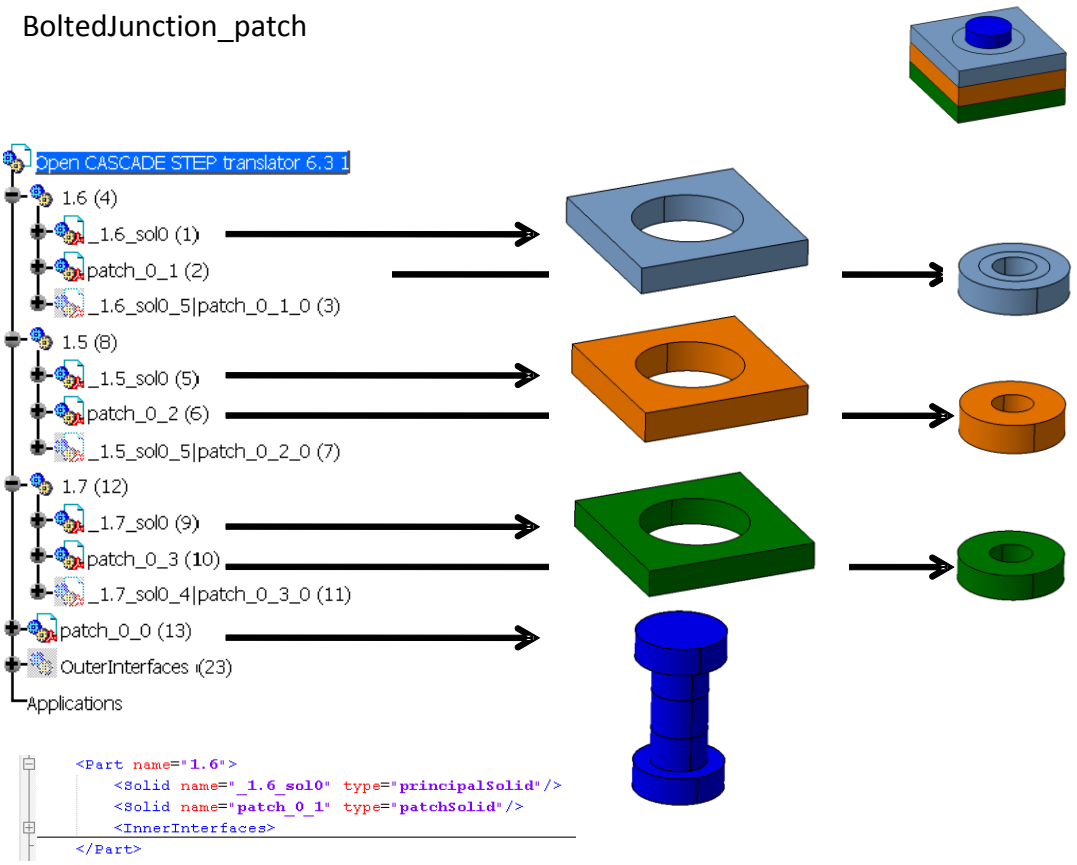


Figure D.2: Illustration of the STEP export of a Bolted Junction. Each component containing volume sub-domains is exported as STEP assembly.

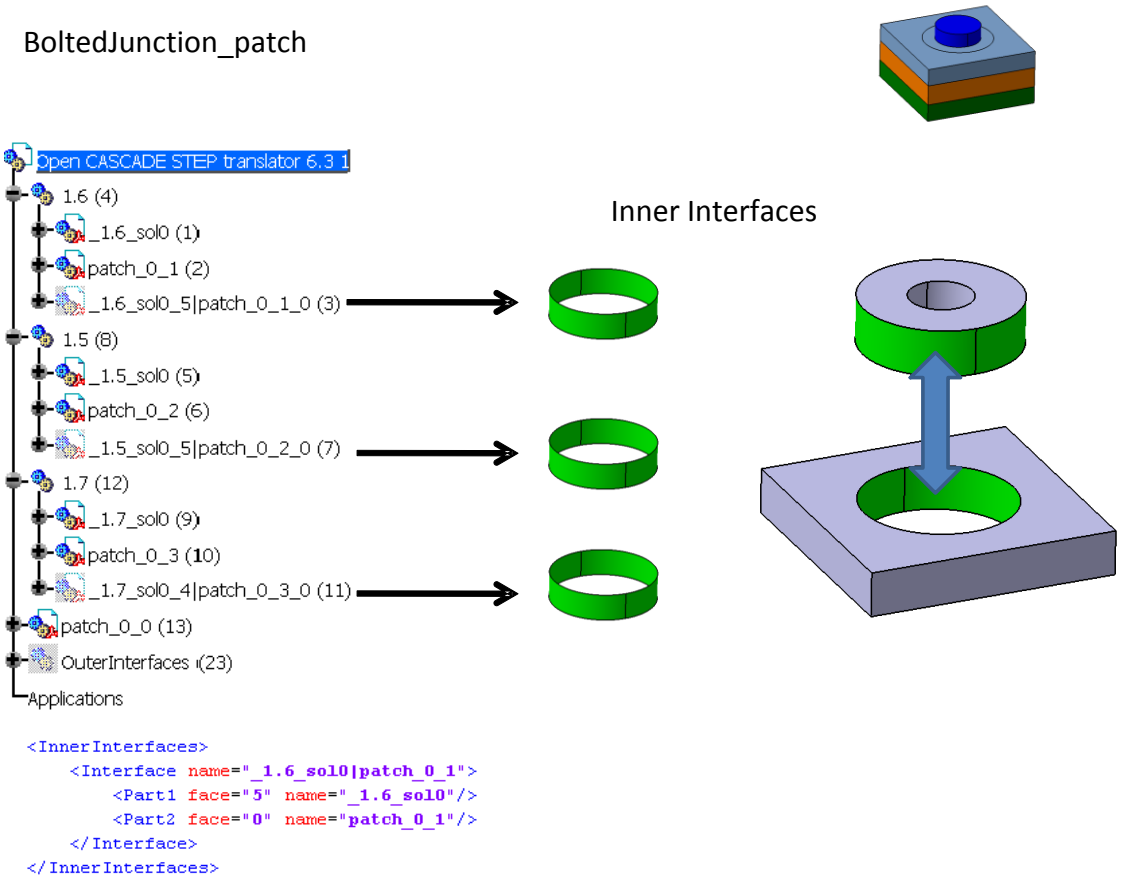
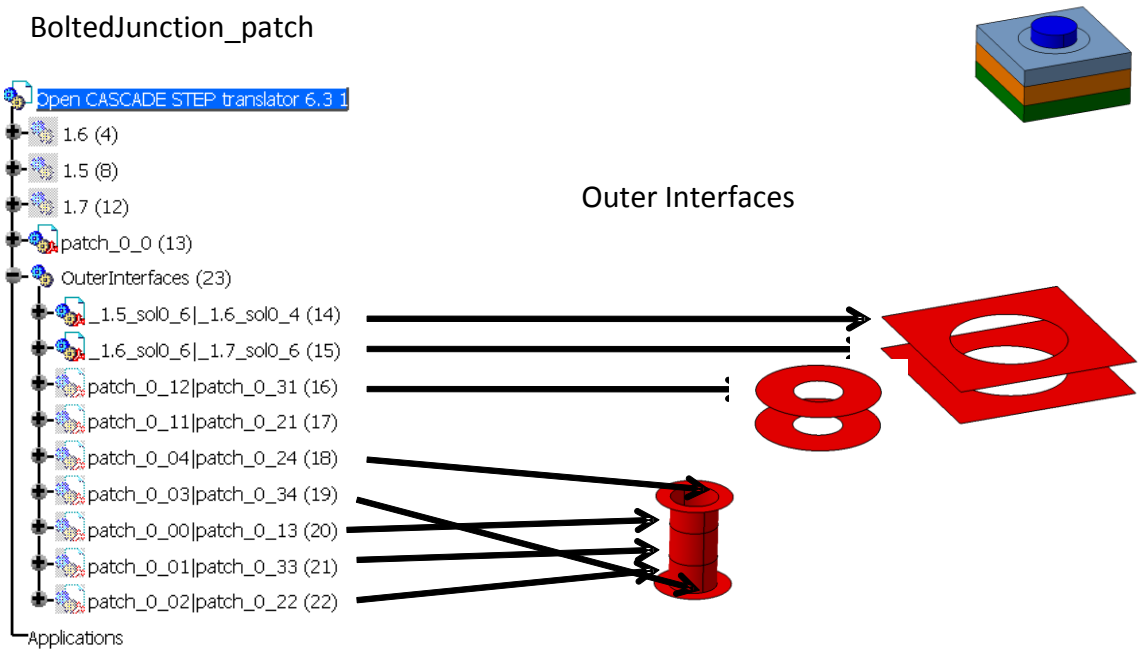


Figure D.3: Illustration of the STEP export of a Bolted Junction. Each inner interface between sub-domains is part of the component assembly.




```
<Interface name="_1.6_sol0|_1.7_sol0">
    <Part1 face="6" name="_1.6_sol0"/>
    <Part2 face="6" name="_1.7_sol0"/>
</Interface>
<interface name="patch_0_1|patch_0_3">
    <Part1 face="2" name="patch_0_1"/>
    <Part2 face="1" name="patch_0_3"/>
</interface>
```

Figure D.4: Illustration of the STEP export of a Bolted Junction. Each outer interface between components is part of the root assembly.

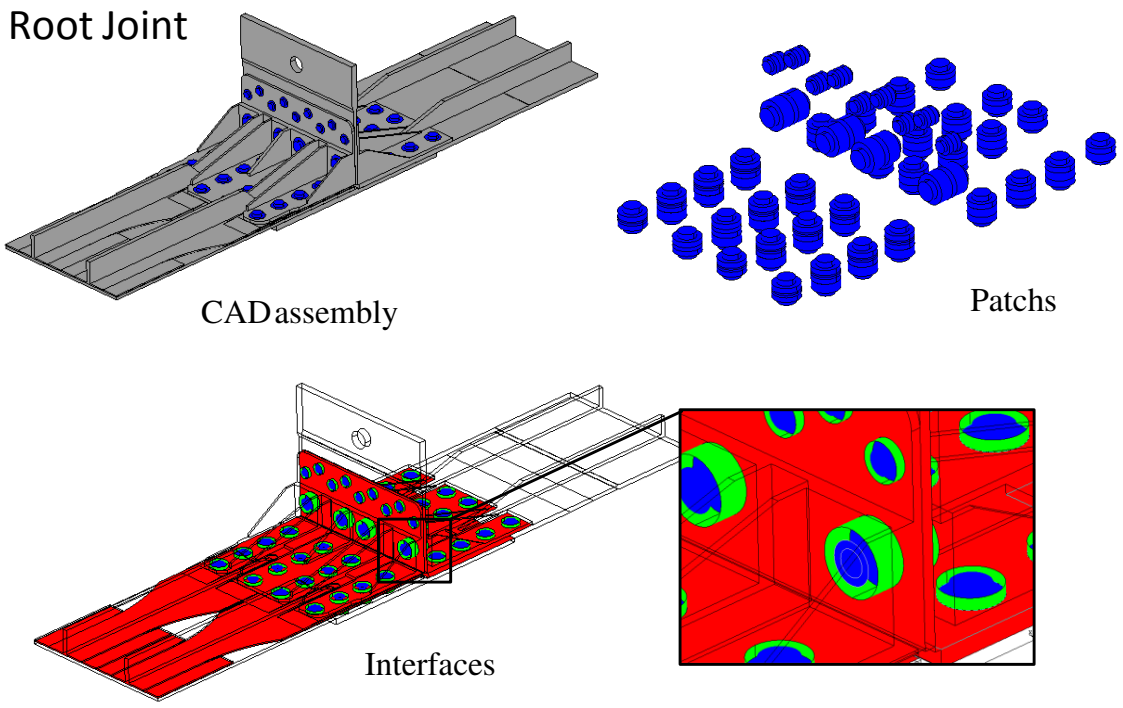


Figure D.5: Illustration of the STEP export of the full Root Joint assembly.